

Segmentação de Imagem PET (Positron Emission Tomography)

Dissertação apresentada para a obtenção do grau de Mestre em Sistemas
e Tecnologias na Informação para a Saúde

Autor

Bruno José Rodrigues de Jesus

Orientador

Francisco Duarte

Instituto Superior de Engenharia de Coimbra

Co-Orientor

Francisco Alves e Ricardo Faustino

Escola Superior de Tecnologia da Saúde de Coimbra

Coimbra, Dezembro 2014

(...) A tese [que aqui se defende] (...) repousa sobre a ideia de que a ciência é uma maneira de perceber e de dar (...) «um sentido» às coisas percebidas. Mas a percepção não se efetua senão por diferenças. Receber uma informação, é necessariamente receber novidades de uma diferença, e toda a percepção (de uma diferença) é limitada por um limiar.

(...) Segue-se que o que nós, homens de ciência, podemos perceber será sempre limitada por um limiar.

G.Bateson (1984) La nature et la pensée, Paris Seuil, pp. 31-35

Agradecimentos

Primeiro que tudo quero agradecer a todos que ao longo destes anos fizeram parte do meu crescimento profissional e pessoal, desde família amigos, colegas de curso também não esquecendo os professores que se cruzaram comigo durante o meu percurso académico.

Segundo agradeço ao Mestre Ricardo Faustino da instituição ICNAS pela brilhante ideia que deu para levar à concretização deste projeto.

Agradeço ao meu orientador Mestre Francisco Duarte do ISEC, pela toda força, ideias que ajudaram a superar dificuldades e sobre tudo a disponibilidade que me dispensou durante este projeto.

Quero agradecer aos meus co-orientadores Doutor Francisco Alves e Mestre Ricardo Faustino da instituição ICNAS pelo material disponibilizado e também pelo tempo que me disponibilizaram.

Por fim quero agradecer ao Mestre Fábio Nery da instituição ICNAS, pelos conselhos e que levou este projeto ainda mais adiante do que previsto inicialmente.

Resumo

O presente relatório é o trabalho desenvolvido no âmbito da unidade curricular de projeto, do Mestrado em Sistemas e Tecnologias de Informação para a Saúde, com a finalidade de demonstrar a concretização dos objetivos e desafios inicialmente propostos e atingidos além.

O objetivo deste trabalho é desenvolver um algoritmo capaz de identificar regiões SUV diferenciadas em PET.

Pretende-se que a segmentação seja concisa de forma a poder ser utilizada em qualquer outra área de estudos científicos, de doenças neste caso.

Ultimamente a informática na área da imagiologia, avançou sentindo cada vez mais necessidades de tornar os diagnósticos mais precoces.

Nas segmentações de imagem os especialistas desempenham o trabalho de forma manual, podendo esta demorar muito e variar com o especialista responsável, para isso procedemos à construção de segmentação de volumes de PET em imagem humana.

De uma forma geral, este algoritmo pretende oferecer aspetos positivos, designadamente que possa vir a ter a possibilidade de adaptação na área da imagiologia.

Abstract

This report is the work done within the course of the project, the Master in Information Systems and Technologies for Health, in order to demonstrate the achievement of objectives and challenges initially proposed and achieved beyond.

The objective of this work is to develop an algorithm capable of identifying SUV regions differentiated in PET.

It is intended that the segmentation is accurate so that it can be used in any other field of scientific studies of disease in this case.

Lately the computer in the field of imaging, advanced feeling increasingly needs to make earlier diagnosis.

In the image segmentation experts play the manual way of working, which may be long and vary with the responsible expert for that proceeded to the construction of targeting PET volumes in human image.

In general, this algorithm aims to offer positive aspects, such as may be possible to adapt the imaging area.

Palavras-Chave

Processamento de imagem, segmentação de regiões, tomografia por emissão de pósitrons.

Key-words

Image processing, regions of segmentation, positron emission tomography.

Índice

Capítulo 1- Introdução	13
1.1. Enquadramento.....	13
1.2. Necessidade da Presente Investigação	15
1.3. Objetivos e Metodologia	16
1.4. Organização do Trabalho	17
Capítulo 2 – Segmentação de Imagem	18
Capítulo 3 – Segmentação de imagem (fantomas) TAC em 2D.....	26
Capítulo 4 – Segmentação de imagem (fantomas) PET em 2D.....	30
Capítulo 5 – Segmentação imagem (humana) PET em 2D	37
Capítulo 6 – Construção de Volume imagens (fantomas) PET 3D.....	44
Capítulo 7 – Construção de Volume imagens (humana) PET em 3D	51
Capítulo 8 – Conclusões gerais e Trabalhos Futuros.....	58
8.1. Síntese do Trabalho e Conclusões Gerais	58
8.2. Prosseguimento de Trabalhos Futuros	59
8.3. Considerações Finais.....	60
Referências Bibliográficas	61
Anexos	62

Índice de Figuras

Figura 1: Planos de seção	16
Figura 2: Exemplo de fantoma PET	19
Figura 3: Valores UINT16.....	20
Figura 4: Valores UINT8.....	20
Figura 5: Exemplo de vizinhança	22
Figura 6: Pixels de um plano uint8.....	22
Figura 7: Gráfico da função MTF.....	23
Figura 8: Fluxo da função MTF	24
Figura 9: Fluxograma do algoritmo.....	25
Figura 10: Plano nº 10 original int8 TAC.....	26
Figura 11: Objetos diferenciados em plano nº10 de imagem TAC	27
Figura 12: Contraste objetos TAC plano nº10.....	28
Figura 13: Valor médio dos objetos TAC no plano nº 10.	28
Figura 14: Tonalidade plano nº10 valor médio dos objetos TAC uint8.....	29
Figura 15: Plano Fantoma inicialmente.....	30
Figura 16: Plano nº45 original int16.....	31
Figura 17: Plano original nº45 em uint8.....	32
Figura 18: Valor médio dos objetos diferenciados do plano nº45 em uint8.....	33
Figura 19: Valor do melhor contraste no limiar do plano nº45 em uint8.....	33
Figura 20: Identificação de localização dos objetos no plano nº45 em uint8.....	34
Figura 21: Intensidade do melhor contraste dos objetos do plano nº45 em uint8.....	35
Figura 22: Intensidade do valor médio dos objetos do plano nº45 em uint8.....	36
Figura 23: Plano original uint8 nº 174.....	37
Figura 24: Representação tridimensional do plano nº174 em uint8.....	37
Figura 25: Representação tridimensional do plano nº 174 original int16	38
Figura 26: Barra de progresso do plano.....	38
Figura 27: Representação tridimensional dos objetos encontrados no plano nº 174 em uint8	39
Figura 28: Representação tridimensional do melhor contraste uint8 no limiar.....	40
Figura 29: Area, número e contorno de objeto.....	41
Figura 30: Objeto de elevado contraste plano nº174	42
Figura 31: Forma e intensidade dos objetos segmentados em uint8	43
Figura 32: Volume fantoma inicial em uint8	44
Figura 33: Corte axial nº 45 volume original fantasmas.....	45
Figura 34: Barras de progresso.....	46
Figura 35: Volume fantasmas segmentado uint8.....	47
Figura 36: Corte axial nº45 volume de fantasmas segmentado	48
Figura 37: Objetos fantasmas pausa do plano nº45	49
Figura 38: Exclusão de objetos fantasmas pausa limiar 109	49
Figura 39: Plano fantoma 45 em pausa	50
Figura 40: Volume inicial uint8.....	51
Figura 41: Corte axial nº 174 no volume original	52
Figura 42: Volume de imagens segmentado em uint8	53
Figura 43: Corte axial nº174 no volume segmentado.....	54
Figura 44: Pausa no plano nº 174	55
Figura 45: Pausa no limiar 128 do volume segmentado.....	56
Figura 46: Objeto detetado na pausa do plano nº 174	57

Índice de Quadros

Quadro 1: Listagem de valores dos objetos do plano nº45 em uint8 e uint16	34
Quadro 2: Valores do objeto de elevado contraste no plano 174 humano	41

Simbologia

C – Contraste do objeto;

CF- Contraste Final;

MO – Média dos pixéis objeto;

MV – Média dos pixéis vizinhança do objeto;

Fuint8 – Formula conversão UINT8;

D_uint16 – *Double* do *pixel* UINT16;

Max_uint16 – Máximo da matriz UINT16;

Max_uint8 – Máximo da matriz UINT8;

Abreviaturas

ISEC – Instituto Superior de Engenharia de Coimbra

ICNAS – Instituto das Ciências Nucleares Aplicadas à Saúde

PET – *Positron Emission Tomography*

TAC – *Tomography Axial Computerized*

SUV – *Standardized Uptake Value*

2D – Duas Dimensões

3D – Três Dimensões

IDE – *Integrated Development Environment*

DICOM - *Digital Imaging and Communications in Medicine*

INT16 – Suporta os valores de $-32,768$ até $+32,767$

UINT8 – Suporta os valores de 0 até $+255$;

UINT16 – Suporta os valores de 0 até $+65,535$

JPEG - *Joint Photographic Experts Group*

GIF - *Graphics Interchange Format*

TIF - *Tagged Image File*

PNG - *Portable Network Graphics*

IOS – Sistema operacional móvel da *Apple*

Capítulo 1- Introdução

1.1. Enquadramento

Este projeto foi realizado em parceria com o ICNAS, para conclusão do Mestrado de Sistemas e Tecnologias de Informação para a Saúde.

Hoje em dia a segmentação de imagens é vista como uma área extremamente importante em Medicina Nuclear.

A Medicina Nuclear é uma especialidade médica que tem como principais objetivos, o diagnóstico de doenças através de pequenas quantidades radioativas, chamadas radiofármacos.

Os exames da Medicina Nuclear são utilizados praticamente em todos os campos da intervenção médica, nomeadamente em oncologia.

Estes exames da Medicina Nuclear também são estendidos a outros campos como cardiologia, endocrinologia, ortopedia, reumatologia, nefro-urologia, gastroenterologia, cirurgia, neurologia, etc.

Para a variedade de campos assinalados, o diagnóstico destes exames possibilita a avaliação funcional e metabólica dos diversos órgãos do sistema do corpo humano de uma forma não invasiva. Permite ainda a identificação de alterações no metabolismo antes de serem visíveis, alterações a nível morfológico constitui uma vantagem significativa, visto que possibilita a deteção de doenças antes de existir sinais morfológicos evidentes. Logo isto só é possível porque a Medicina Nuclear permite avaliar o funcionamento dos órgãos e não apenas a sua estrutura ou morfologia.

Devido a um grande avanço nas ciências biomédicas acelerou-se a introdução de novas tecnologias de imagiologia nos últimos anos, de forma mais precisa e eficiente torna-se um fator para o sucesso da investigação.

A tecnologia PET é uma técnica de imagiologia usada em Medicina Nuclear que fornece dados necessários para o mapeamento da distribuição de um radiofármaco injetado no organismo de um doente previamente. Podemos através deste mapeamento obter informação de regiões ou órgãos que possam estar fora da normalidade.

Esta técnica de imagiologia nuclear tem como base, o uso de emissores de positrões como radiofármacos.

Existem vários tipos de radiofármacos utilizados na PET, para sabermos a qual o tipo de diagnóstico de doença ajudam na sua detecção, vamos mencionar alguns deles (2014, Informação para profissionais de saúde);

- PET *¹⁸F-FDG* é um derivado da glicose, hoje em dia é o mais utilizado, com maior aplicação nos campos de oncologia, neurologia e cardiologia. A sua biodistribuição vai incidir particularmente a nível cerebral, pulmonar, hepático e cardíaco.
- PET *¹¹C-PiB* está associado ao diagnóstico da doença de *Alzheimer*.
- PET *¹⁸F-FDG + ¹¹C-PiB* é o estudo da sequência do *¹⁸F-FDG + ¹¹C-PiB*.
- PET *⁶⁸Ga-DOTANOC* permite o diagnóstico tumores neuroendócrinos bem diferenciados.
- PET *¹¹C-Raclopride* permite a observação e análise destes neuroreceptores.
- PET *¹¹C-Flumazenil* permite-nos o diagnóstico para a doença de *Alzheimer*, doença de *Huntington*, distúrbios da ansiedade, síndrome de *Angelman*, encefalopatia hepática, esquizofrenia e epilepsia.
- PET *¹⁸F-NaF* é utilizado para diagnóstico da imagem do osso.
- PET *¹⁸F-Colina* é utilizado no diagnóstico da próstata.
- PET *¹¹C-Metionina* é utilizado no estudo do metabolismo dos aminoácidos.

Uma característica importante na PET é a resolução espacial, existindo fatores que a afetam, existindo uma limitação física, assim como melhor sensibilidade, a resolução espacial não diminui com o aumento da distância do objeto e a correção de atenuação é fácil e precisa.

Contudo técnica PET apenas pode ser realizada em centros especializados (como por exemplo o ICNAS), devido a questões regulamentares, de produção e também de manuseamento dos radiofármacos que emprega equipamentos muito caros (como por exemplo o ciclotrão).

1.2. Necessidade da Presente Investigação

A presente investigação nasceu da necessidade de desenvolver um sistema inovador capaz de encontrar regiões de SUV (1) diferenciadas, classificando as e validando as num plano bidimensional em imagens PET (2014,SUV).

$$SUV = Activity\ Concentration\ in\ ROI / \{ Injected\ Dose | Body\ Weight \} \quad (1)$$

Através de algoritmos como este podem dar um avanço automático da imagem PET, para o desenvolvimento da Medicina Nuclear, bem como na investigação de novos radiofármacos para diagnóstico e tratamento de doentes.

Com o desenvolvimento de novos equipamentos para diagnóstico no futuro, será mais promissor as imagens adquiridas, permitindo melhor resolução.

A Medicina Nuclear é uma medicina personalizada e como está na ordem do dia, renunciando o desenvolvimento futuro de cuidados médicos radicalmente diferentes até aos dias de hoje praticados.

Podemos ver o futuro da Medicina Nuclear como um intuito de melhorar o diagnóstico quer na terapêutica como no desenvolvimento científico nesta área.

Penso que dado o avanço tecnológico seria necessário presentemente um sistema capaz de segmentar um volume de imagens, capaz de identificar diferentes regiões como por exemplo órgãos e calcular exatamente o volume médio desses objetos e o seu contraste.

Contudo existe sempre uma razão para inovar relativamente aos mecanismos de alarme para diagnóstico prévio.

1.3. Objetivos e Metodologia

1.3.1. Principais objectivos

Este projeto vai incidir no desenvolvimento de um *software*, para:

- Regiões diferenciadas em UINT8/UINT16;

Identificar as diferentes regiões num plano bidimensional.

- Média da região do objeto em UINT8/UINT16;

$$MO = \text{Soma dos pixels da região} \div \text{número de pixels} \quad (2)$$

- Contraste da região do objeto em uint8/uint16;

$$C = MO - MV \quad (3)$$

- Listagem *Excel* dos valores do plano 2D;

Exportar tabela do *Matlab* para *Excel*.

- Construção de volume em 3D de planos 2D;

Soma de todos planos indexados numa matriz 3D.

- Corte axial ou transverso;

Divide corpo em duas metades, superior e inferior;

- Corte coronal ou frontal;

Divide corpo em duas metades, anterior e posterior;

- Corte sagital,

Divide corpo em duas metades, esquerda e direita (Anatomia Humana, 2011);

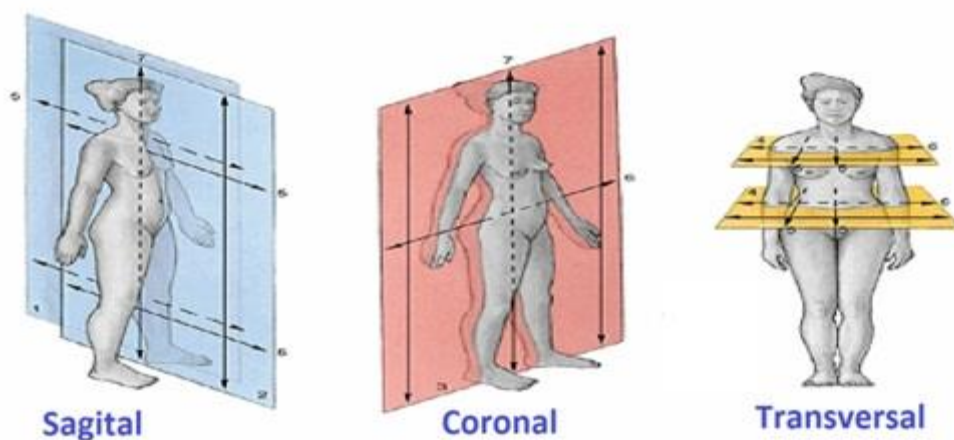


Figura 1: Planos de seção

1.3.2. Metodologia

Foram efetuadas reuniões semanais com o orientador, com seguimento de troca de ideias, correções de acordo com a complexidade do trabalho. Algumas trocas de ideias e partilha de ficheiros foram feitos através da *DropBox* (serviço de armazenamento e partilha de ficheiros, desde que haja acesso a internet).

Este serviço de partilha para comunicação entre aluno e orientador foi uma das formas de nos encontrar sincronizados.

Durante o desenvolvimento do projeto foram realizadas algumas reuniões com o Doutor Francisco Alves e Mestre Ricardo Faustino da Instituição ICNAS, com a finalidade de mostrar resultados do objetivo inicialmente proposto, assim como sugerir novos desafios atingir.

1.4. Organização do Trabalho

O presente trabalho encontrasse dividido em 8 capítulos, onde ao longo destes se abordam inicialmente os conceitos básicos de Medicina Nuclear como aspetos necessários à realização deste trabalho, assim como testes ao algoritmo desenvolvido em diferentes imagens médicas.

Mais concretamente o primeiro capítulo serve de introdução ao tema abordado, como os objetivos pretendidos e a metodologia seguida para conseguir atingir o fim pretendido.

O segundo capítulo define o algoritmo de desenvolvimento do projeto de segmentação de imagem PET.

O terceiro capítulo diz respeito a uma aplicação desenvolvida para segmentação de imagens fantomas TAC, em planos 2D.

O quarto e quinto capítulo apresentam uma aplicação desenvolvida para segmentação de imagens fantomas e humanas PET, em planos 2D, que permite visualizar planos em 3D, plano segmentado, quantidade de objetos no plano e ainda valor médio dos objetos e contraste de cada objeto independentemente.

O sexto e sétimo capítulo são o complemento de todas as aplicações desenvolvidas inicialmente, mostram a construção do volume inicialmente e depois de ser segmentado com a possibilidade de ver em vídeo todos os planos em 2D e 3D a intensidade do contraste e por fim o volume segmentado completo em 3D eliminando regiões de baixo contraste.

Este relatório apresenta no oitavo capítulo uma breve conclusão com comentários pessoais e ainda alguns aspetos para trabalhos futuros.

Capítulo 2 – Segmentação de Imagem

O *software* de segmentação de imagens PET foi desenvolvido em ambiente *Matlab R2011a*, mas no final foi migrado para o *Matlab R2014a* para compilar um ficheiro executável mais elaborado.

Esta escolha deveu-se a algumas razões:

- Existe maior aptidão nesta *IDE* do que comparativamente a outros (*Eclipse, NetBeans, Visual Studio, Adobe Flash Builder...*).
- Temos várias funções predefinidas em *toolboxes* que nos facilitam muito a análise e o processamento de imagens.
- É um sistema interativo que nos devolve rapidamente a informação da matriz, resolvendo muitos problemas numéricos numa fração de tempo reduzida.
- Por último podemos dizer que devido à função *guide* permite criar uma *interface* rápida e intuitiva.

Para elaborar este relatório foram utilizadas também as seguintes ferramentas de apoio;

- SNAGIT EDITOR – Software de edição e captura de imagem;
- VISUSTIN – Software de elaborar diagramas de fluxo;
- MICROSOFT VISIO 2007 – Software de criação de diagramas de fluxo;
- MICROSOFT EXCEL 2013 – Software criar tabela de dados de objetos;
- ADVANCED GRAPHER – Software de criação de gráficos matemáticos.

Primeiro que tudo este *software*, foi desenvolvido e testado numa máquina com um sistema operativo de 32 *bits*. Referencio esta mensagem visto que se este for testado em máquinas de 64 *bits* tornem o tempo de segmentação para menos de metade.

O problema inicial para o desenvolvimento deste algoritmo era encontrar o objeto de interesse nos fantasmas, perceber a sua essência a nível do *pixel*.

Começamos por perceber que os ditos fantasmas não eram mais do que tubinhos de diferentes alturas, que continham materiais equivalentes ao tecido e com material radioativo.

A figura nº 2 mostra um exemplo de um fantoma de PET vazio sem material radioativo (2014, Fantomas).



Figura 2: Exemplo de fantoma PET

O porquê a nível de *pixel*, visto que a imagem inicial do fantoma ou mesmo imagem humana todas elas em planos bidimensionais estão no formato DICOM, assumindo valores de INT16.

Partindo que seria muito difícil percorrer toda a matriz em INT16, seriam muitos limiares a testar, mas fomos avisados que a essência do *pixel* era tudo na imagem médica, logo não podíamos alterar para formato, como JPEG, GIF, PNG, TIF, etc.

Tendo assim a imagem continuar no formato DICOM, sabendo que a imagem original INT16 assume valores entre $-32,768$ até $+32,767$, resolvemos tornar estes valores somente em valores positivos, ou seja para valores de 0 até $+65,535$ UINT16.

O Matlab R2011a, ao fazer a conversão de UINT16 para UINT8, não faz de maneira correta, perdendo essência do *pixel*.

Não alterando o formato DICOM, convertemos através de uma fórmula, a imagem assumir valores entre 0 até $+255$ UINT8, da seguinte forma;

- Obtemos o máximo da matriz UINT16;
- Converte se os valores de UINT16 em *double*;

- Multiplica se valores *double* de UINT16 pelo máximo de UINT8, ou seja 255;
- Dividindo pelo máximo da matriz de UINT16;

Observe a seguinte fórmula (4) para conversão para a gama de valores UINT8;

$$F_{uint8} = D_{uint16} \times Max_{uint8} \div Max_{uint16} \quad (4)$$

Para comprovar que conseguimos obter uma imagem DICOM em UINT8 sem perder as características do *pixel*, observe o exemplo abaixo;

14312	13056	13915	15694
13111	10848	11500	14063
14013	12156	12901	15312

Figura 3: Valores UINT16

A matriz UINT16 tem máximo de 23095, o valor do *pixel* em causa é 10848.000 em *double*, o máximo da matriz de UINT8 é 255 logo, $10848.000 \times 255 / 23095 = 119,776575$ arredondando o valor de *double* para UINT8 é necessariamente 120, conforme se encontra na figura abaixo.

158	144	154	173
145	120	127	155
155	134	142	169

Figura 4: Valores UINT8

Todos os cálculos são feitos em *double* para não perder valores que possam ser mal arredondados.

Ao converter todas as matrizes inicialmente para UINT8, estamos otimizar o tempo de execução será mais fácil percorrer limiares de 0 até 255 (UINT8) do que 0 até 65535 (UINT16). Logo vamos percorrer uma matriz de 144 linhas e 144 colunas até 255 limiares para encontrar todos os objetos do plano.

O processo de segmentação baseia se na partição da imagem em dois segmentos distintos, pela escolha de um limiar de binarização conveniente para extração de características importantes para a classificação dos objetos.

A binarização é considerada uma das técnicas mais básicas de processamento de imagem, sendo um método intuitivo de seleção de um limiar, que se baseia na divisão do histograma da gama UINT8, pela seleção do nível de cinzento que corresponde ao vale bimodal que nos separa os objetos do fundo da imagem.

Seja, seleciona se um limiar de binarização para segmentar a imagem em *pixels* pertencentes ao objeto e a 0 ou a 1, objetos pertencentes ao fundo da imagem.

Podemos considerar que o resultado da binarização duma imagem $f(x, y)$ através de um limiar t é dada pela seguinte equação (5);

$$f(x, y) = \begin{cases} b0 & \text{se } f(x, y) < t \\ b1 & \text{se } f(x, y) \geq t \end{cases} \quad (5)$$

Após concluirmos a binarização em todos os limiares e obtermos uma matriz com os elementos pertencentes ao objeto e fundo de imagem, iniciaremos o tratamento desta imagem binarizada para podermos proceder ao cálculo da média de objetos assim como média da vizinhança para todos eles diferenciadamente.

O tratamento da imagem binarizada consiste em colocar a primeira e última linha e primeira e última coluna a zeros, como podemos verificar um pequeno exemplo (6).

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 123 & & & & & 0 \\ 0 & & & & & & 0 \\ 0 & & & & & & 0 \\ 0 & & & & & & 0 \\ 0 & & & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \quad (6)$$

Colocamos todas as extremidades da matriz binarizada a zeros, para poder percorrer os vizinhos de um objeto, se o *pixel* da primeira linha, da primeira coluna, ou seja $f(1,1)$ for um objeto, não conseguiremos detetar vizinhos,

A forma para encontrar todos os diferenciados *pixels* do objetos, foi realizada através do conceito de vizinhança entre os *pixels*. Podemos observar na figura abaixo um pequeno exemplo, a posição inicial é a verde (X;Y), sendo este o pilar vamos verificar vizinhança e guardar as coordenadas de todos os elementos que possam pertencer ao objeto.

	X-1;Y-1	X-1;Y	X-1;Y+1	
	X;Y-1	X;Y	X;Y+1	
	X+1;Y-1	X+1;Y	X+1;Y+1	

Figura 5: Exemplo de vizinhança

Tendo a imagem binarizada tratada, vamos tratar região por região diferenciadamente, em cada limiar será calculado a média dos valores do objeto, assim como o cálculo da média da vizinhança do objeto e guardando sempre a melhor diferença entre as duas médias.

A figura abaixo mostra-nos o valor dos *pixels* em UINT8, estes valores inicialmente estavam em INT16, através da fórmula já referida neste relatório.

Imaginamos que para um dado limiar os valores com quadrado a encarnado são supostamente os *pixels* do objeto, e a azul são supostamente os seus vizinhos. Estou dizendo supostamente porque tudo depende do limiar em causa e da atenuação da nossa função MTF, (ver função MTF em anexos), observe também a figura nº6 (1994, Sensor óptico).

37	116	190	197	159	139	158	191	196	161	120	116	155	189	179	148	137	162	194	201	190	186	198	211	214	207	189	170	168	184	194
38	119	201	226	205	180	174	184	187	170	144	145	173	186	165	132	122	147	181	183	158	144	154	173	190	198	197	191	193	211	222
33	105	184	228	234	199	160	157	178	187	175	171	182	186	174	156	148	166	191	183	145	120	127	155	182	194	201	206	214	229	235
30	93	166	217	228	184	129	126	164	193	188	170	165	177	197	206	197	192	200	189	155	134	142	169	192	197	196	199	204	213	213
27	88	161	204	211	176	124	114	149	186	188	155	134	153	194	219	206	184	184	189	182	177	182	193	198	189	179	176	182	190	193
21	77	153	194	201	187	154	139	156	181	186	154	123	137	176	198	177	143	140	163	187	204	211	208	191	161	137	137	155	174	182
16	65	142	190	198	201	185	164	168	188	196	173	149	158	186	192	156	113	109	138	175	201	216	213	186	143	110	111	142	169	178
10	46	120	183	202	208	199	163	150	174	199	197	185	193	210	206	172	132	124	150	184	207	213	208	189	151	119	120	149	175	183
7	31	97	175	207	214	203	159	126	143	190	211	199	191	198	206	200	183	180	194	209	211	200	189	183	170	155	159	184	205	211
4	20	75	160	213	224	211	165	122	132	180	207	181	150	151	178	202	207	210	212	203	185	170	163	166	177	184	196	222	246	243
3	13	51	126	201	232	220	177	137	143	180	194	157	115	114	149	186	200	202	196	173	144	125	131	152	178	191	196	213	236	241
1	7	32	90	167	213	212	192	170	166	183	190	163	125	116	141	176	196	200	189	161	126	109	122	152	174	167	149	143	159	184
1	4	17	57	128	185	202	200	183	172	184	198	194	176	167	171	184	199	208	202	178	145	128	147	181	186	156	118	98	108	142
0	2	8	33	88	150	188	196	175	153	156	177	198	216	229	219	198	190	194	201	206	195	181	188	205	200	168	126	100	109	145
0	1	3	14	50	111	167	186	167	134	122	142	181	224	252	237	192	155	150	170	203	222	218	211	210	208	190	156	134	148	181

Figura 6: Pixels de um plano uint8

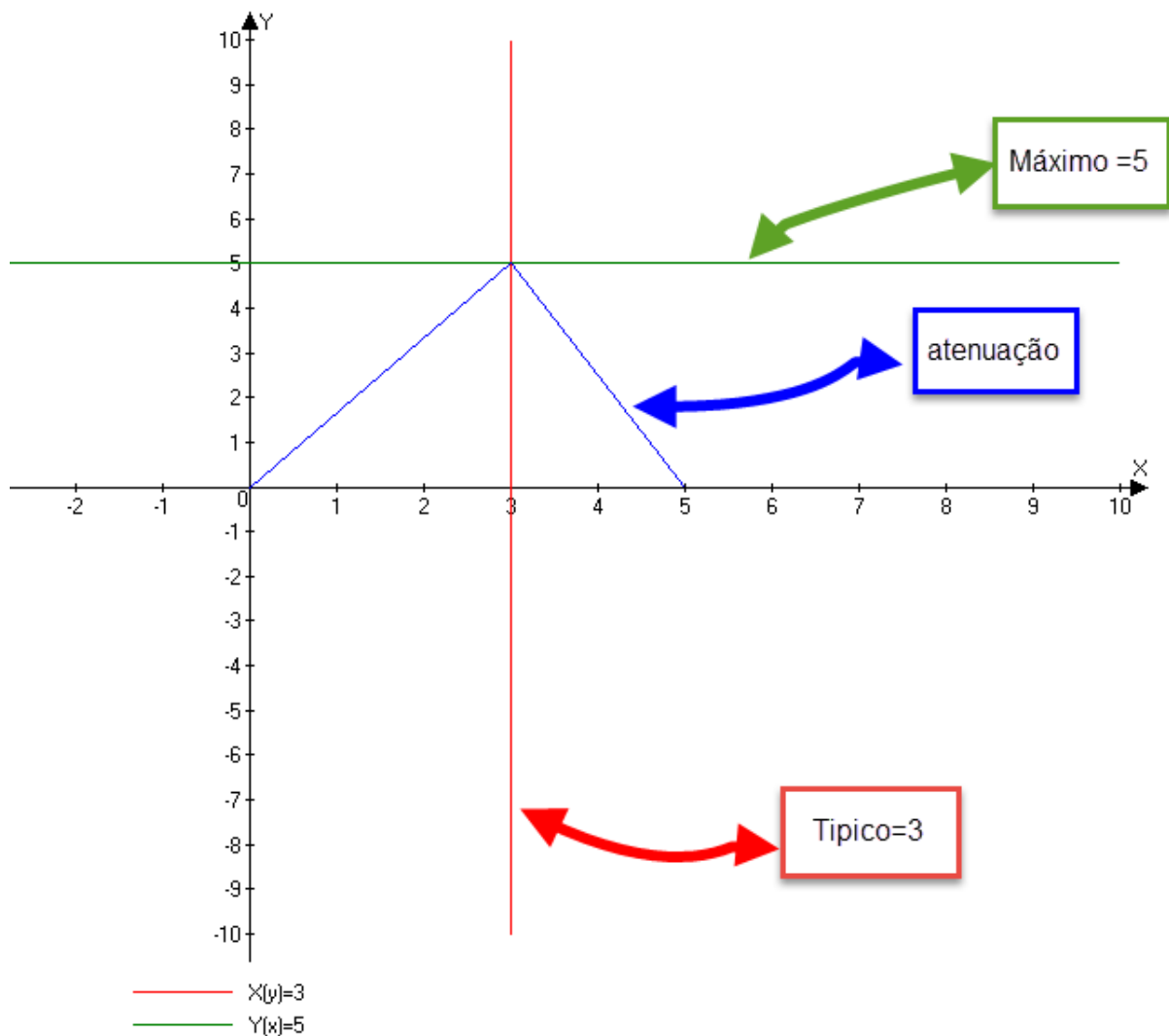


Figura 7: Gráfico da função MTF

Esta função permite-nos delimitar objetos, tem como entrada a área, sendo o número de *pixéis* do objeto e como parâmetro de saída a atenuação.

Os valores de máximo e de típico que definimos na função podem ser ajustados de acordo com a matriz que se esteja a segmentar, estes são os valores que são usados na segmentação de imagens humanas.

O fluxo da função está na figura nº8 esquematizado, mostra-nos como podemos atenuar regiões de grande dimensão para obter melhor percepção dos objetos, visto que em PET, a resolução não é de qualidade, logo temos *pixels*, com informação muito dispersa.

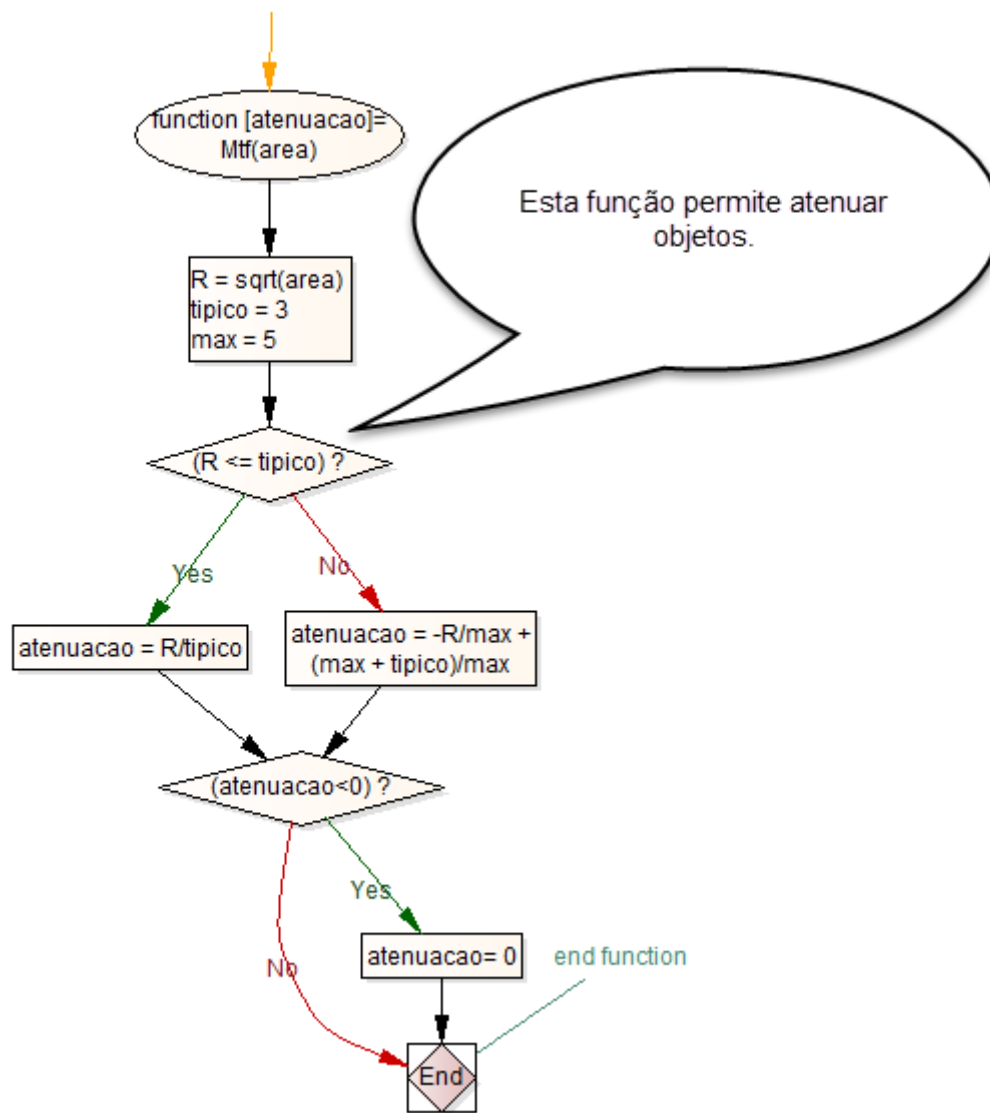


Figura 8: Fluxo da função MTF

Para todos os limiares utilizamos a nossa função de atenuação através das seguintes fórmulas (3) e (6);

$$C = MO - MV \quad (3)$$

$$CF = C \times MTF(\text{número de pixels}) \quad (7)$$

Para percebermos o fluxo do algoritmo podemos observar a figura nº 9.

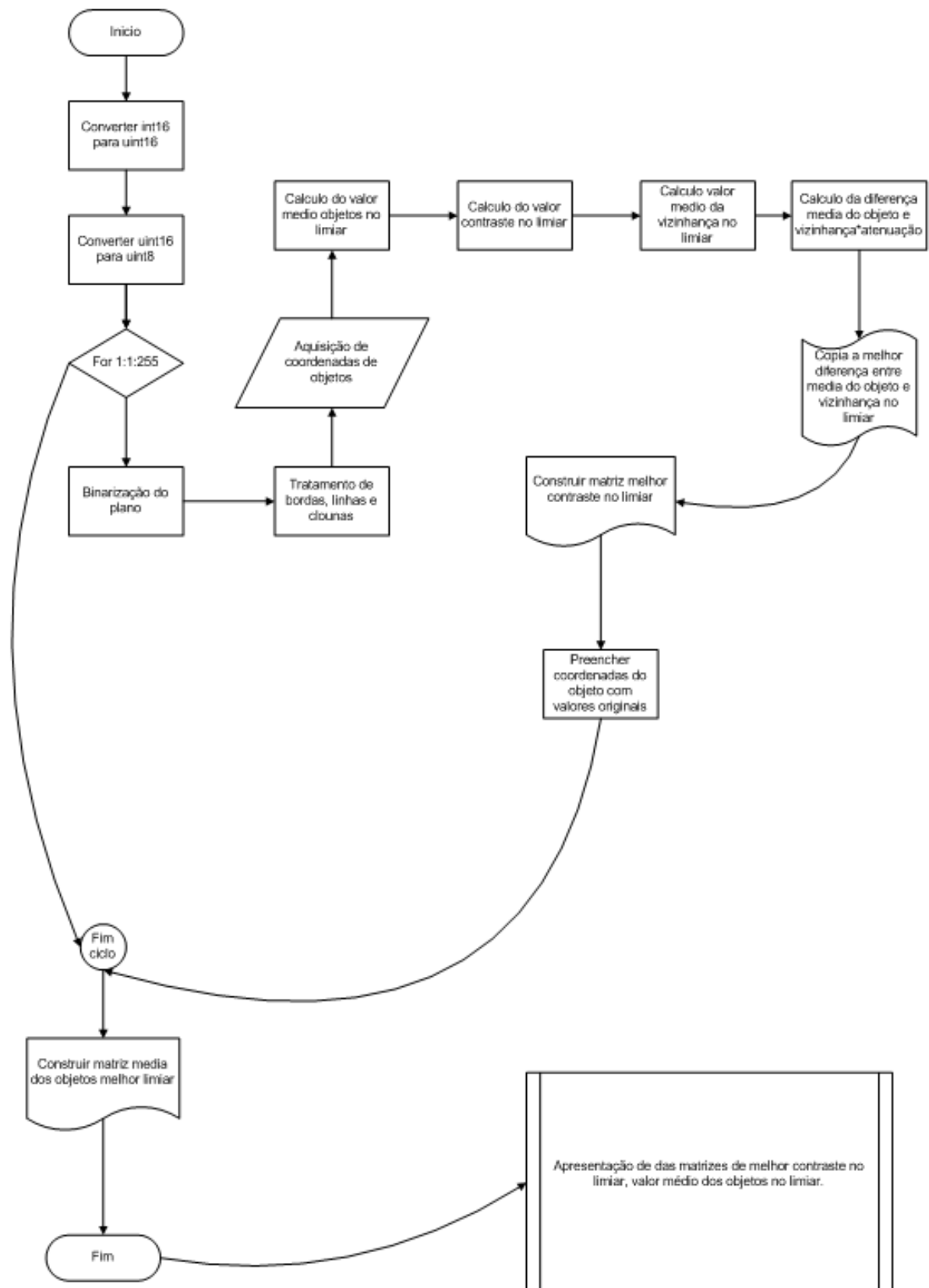


Figura 9: Fluxograma do algoritmo

Capítulo 3 – Segmentação de imagem (fantomas) TAC em 2D

Inicialmente fizemos segmentação de TAC, são muito bem contrastadas, com uma resolução, talvez esse o motivo que levou a fazer segmentação pela tão boa aparência de imagem.

Dado que esta segmentação foi um trabalho, onde tudo assentou, para perceber a essência dos fantasmas dos objetos que teríamos de encontrar.

Como a TAC é muito bem contrastada, apresenta uma matriz quase quatro vezes superior a uma matriz de PET, podemos visualizar a figura abaixo, logo o tempo de segmentação de um plano terá duração de cerca de 15 minutos, visto que estamos a executar numa máquina com sistemas de 32 *bits*.

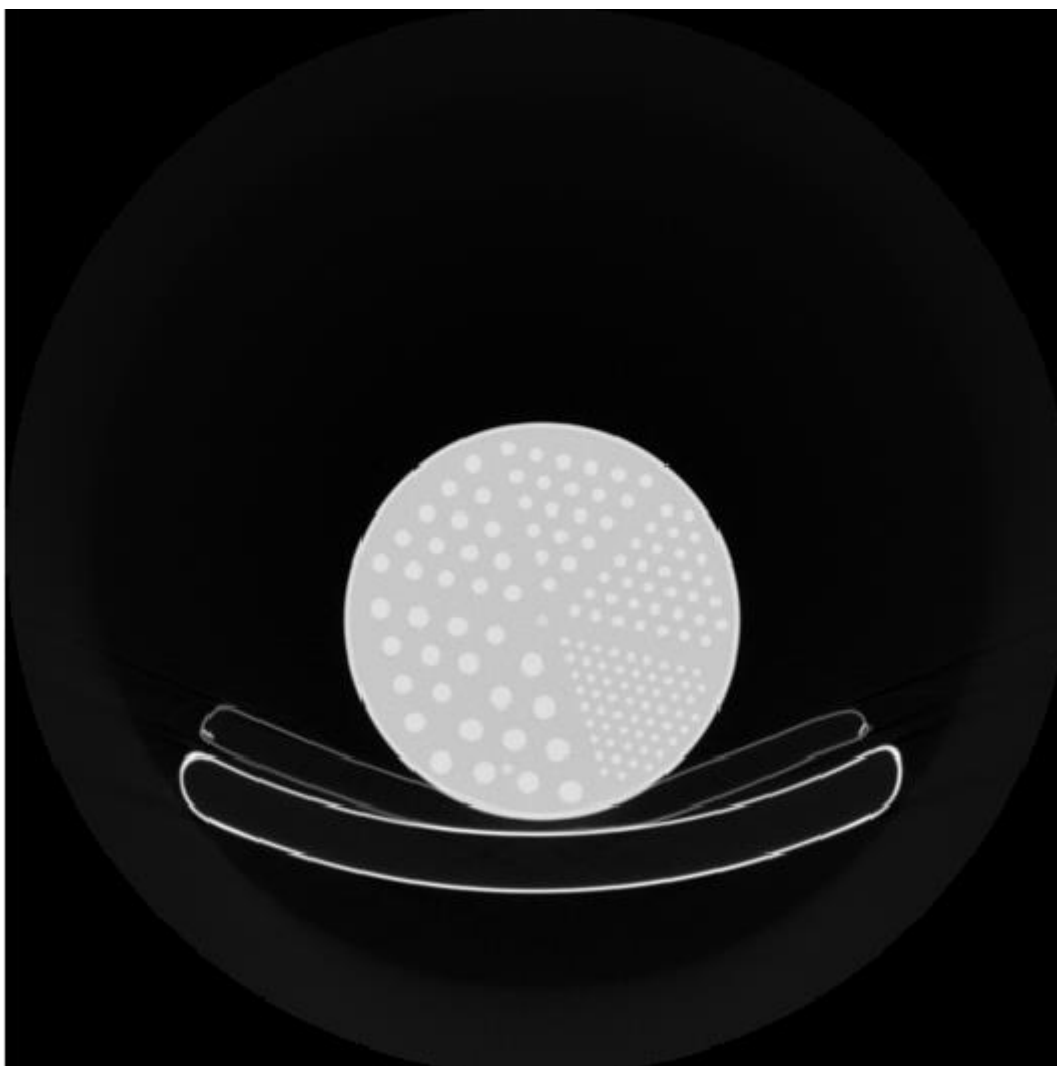


Figura 10: Plano nº 10 original int8 TAC

A figura nº 11 representa todos os objetos diferenciados encontrados num dado plano de imagem TAC.

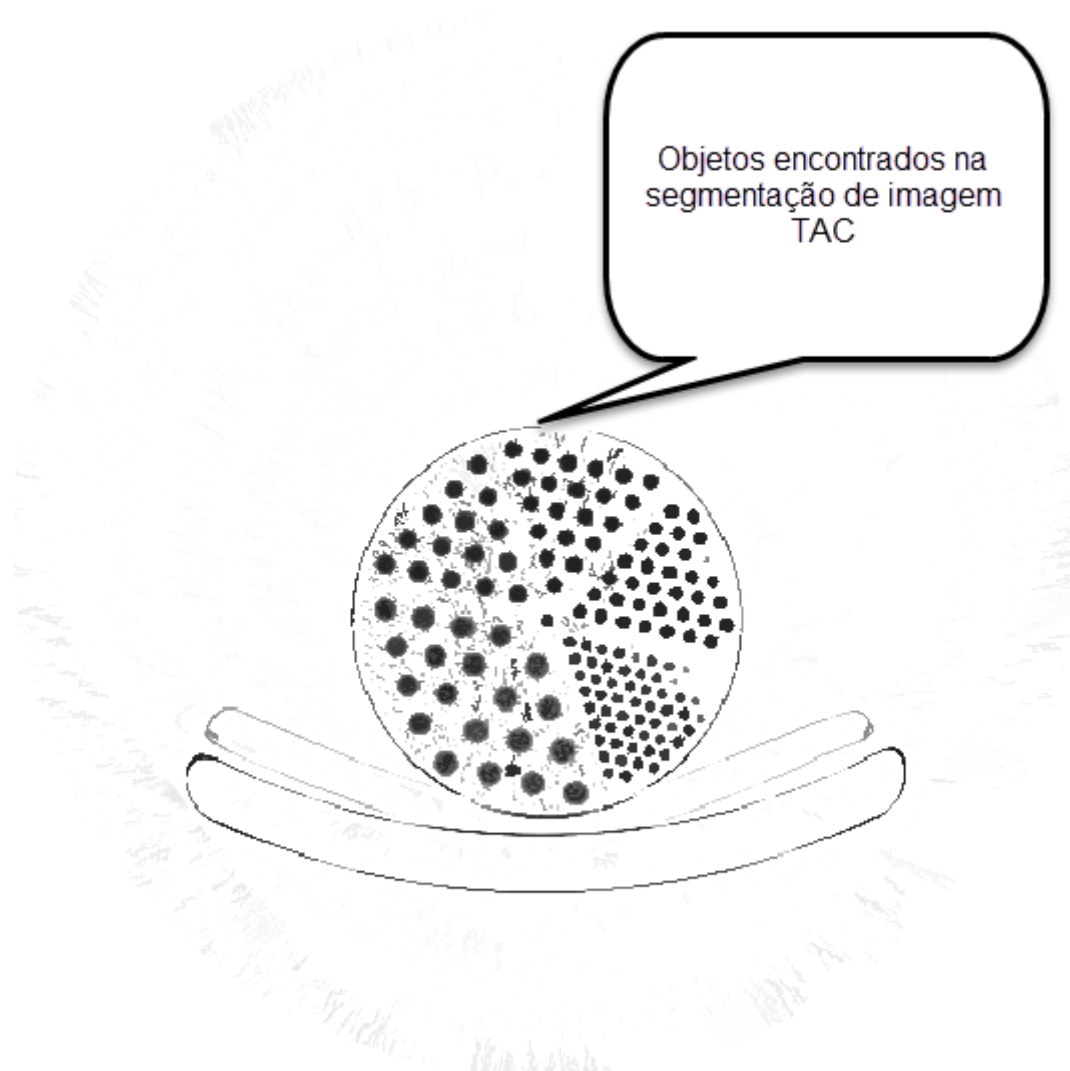


Figura 11: Objetos diferenciados em plano nº10 de imagem TAC

Podemos ver na figura nº 12 o contraste de todos os objetos TAC em 3D encontrados no plano nº10.

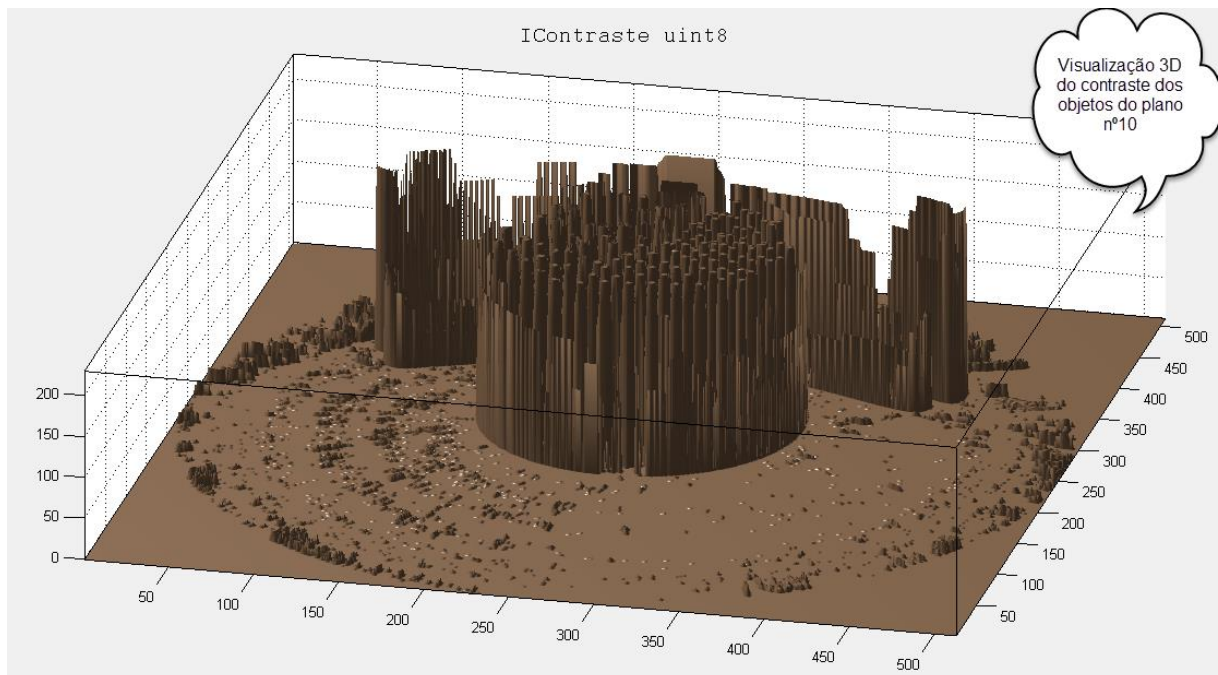


Figura 12: Contraste objetos TAC plano nº10

A figura nº 13 mostra o valor individual da média dos objetos encontrados no plano nº 10 da imagem TAC.

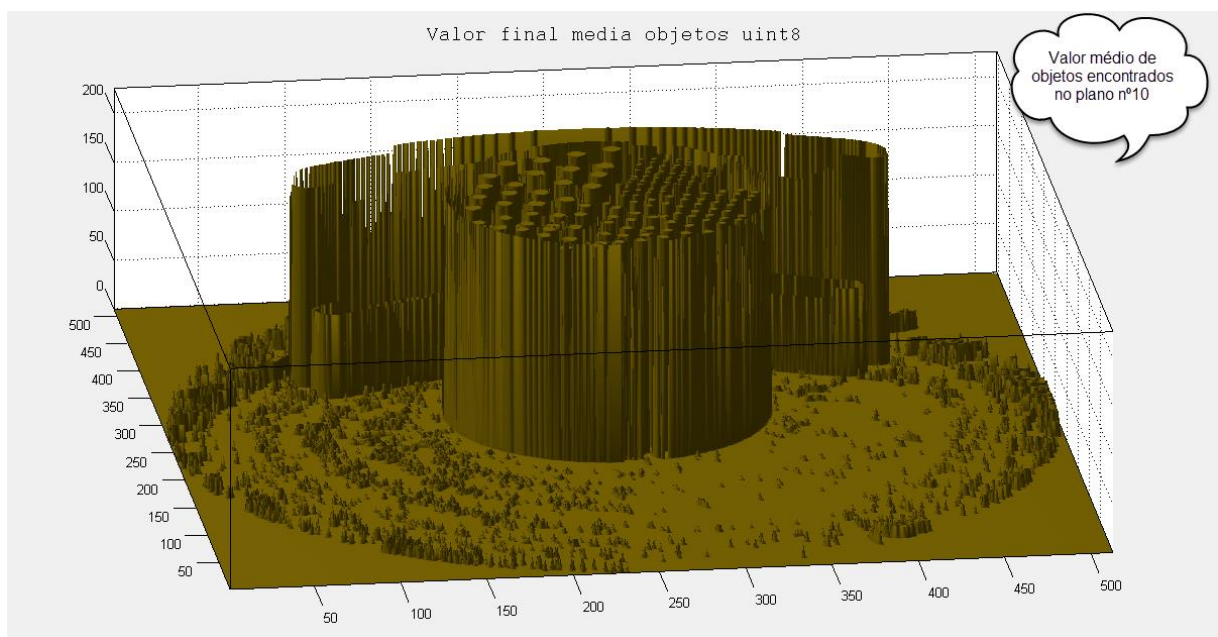


Figura 13: Valor médio dos objetos TAC no plano nº 10.

Na segmentação de TAC por fim a figura nº 14 representa uma imagem com a tonalidade da média dos objetos encontrados no plano nº10.

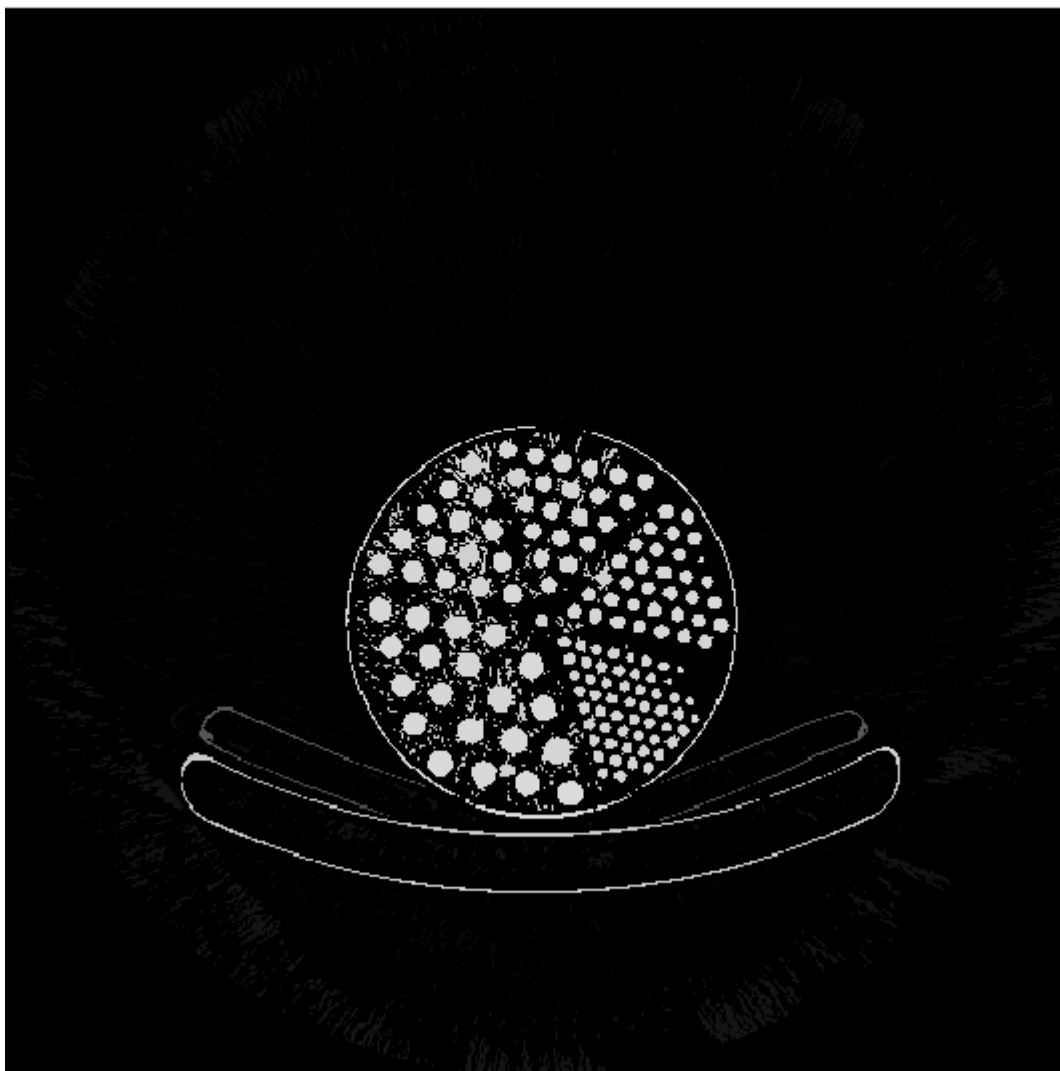


Figura 14: Tonalidade plano nº10 valor médio dos objetos TAC uint8

Capítulo 4 – Segmentação de imagem (fantomas) PET em 2D

A figura nº 15 mostra-nos a uma pequena noção de como é a resolução da imagem PET, neste caso estamos a falar de um plano que vamos tratar.

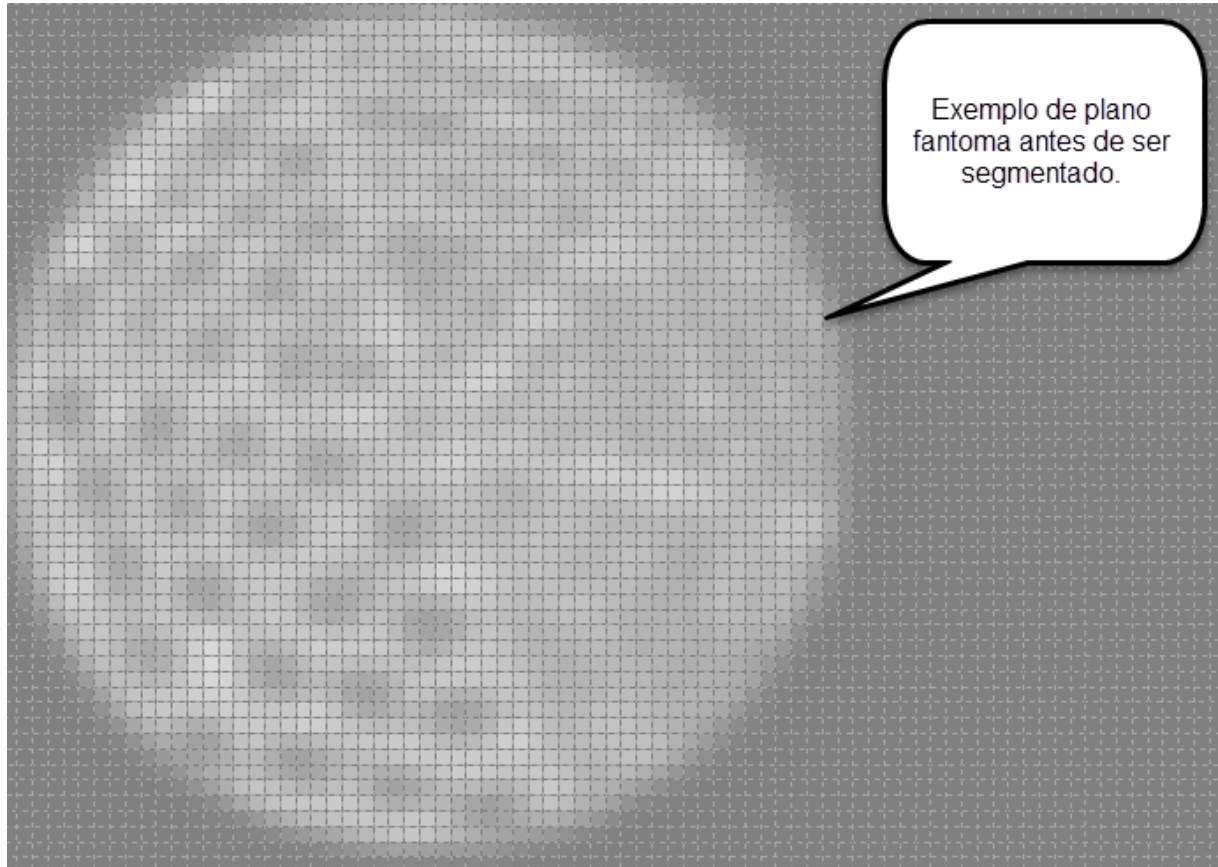


Figura 15: Plano Fantoma inicialmente

Seguindo o algoritmo desenvolvido para imagem TAC, mas com alguns ajustes e parametrizações, aplicamos à imagem PET, figura nº 16 podemos visualizar o plano nº 45 na gama original INT16.

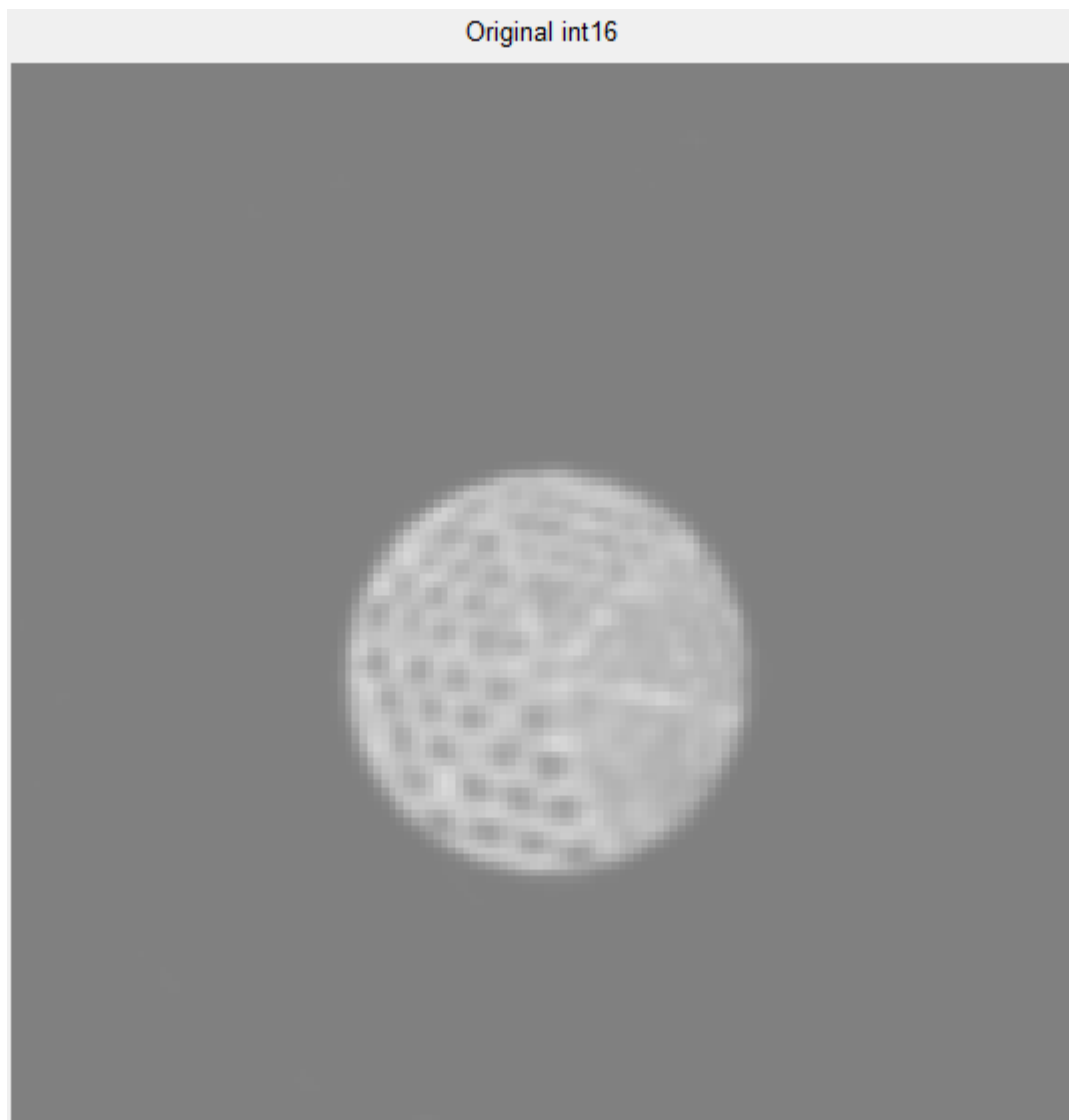


Figura 16: Plano nº45 original int16

Como já referimos neste relatório foi necessário converter a imagem de INT16 para UINT8, vamos visualizar uma imagem na figura nº 17 que nos faz perceber o quanto é mais contrastada a imagem UINT8.

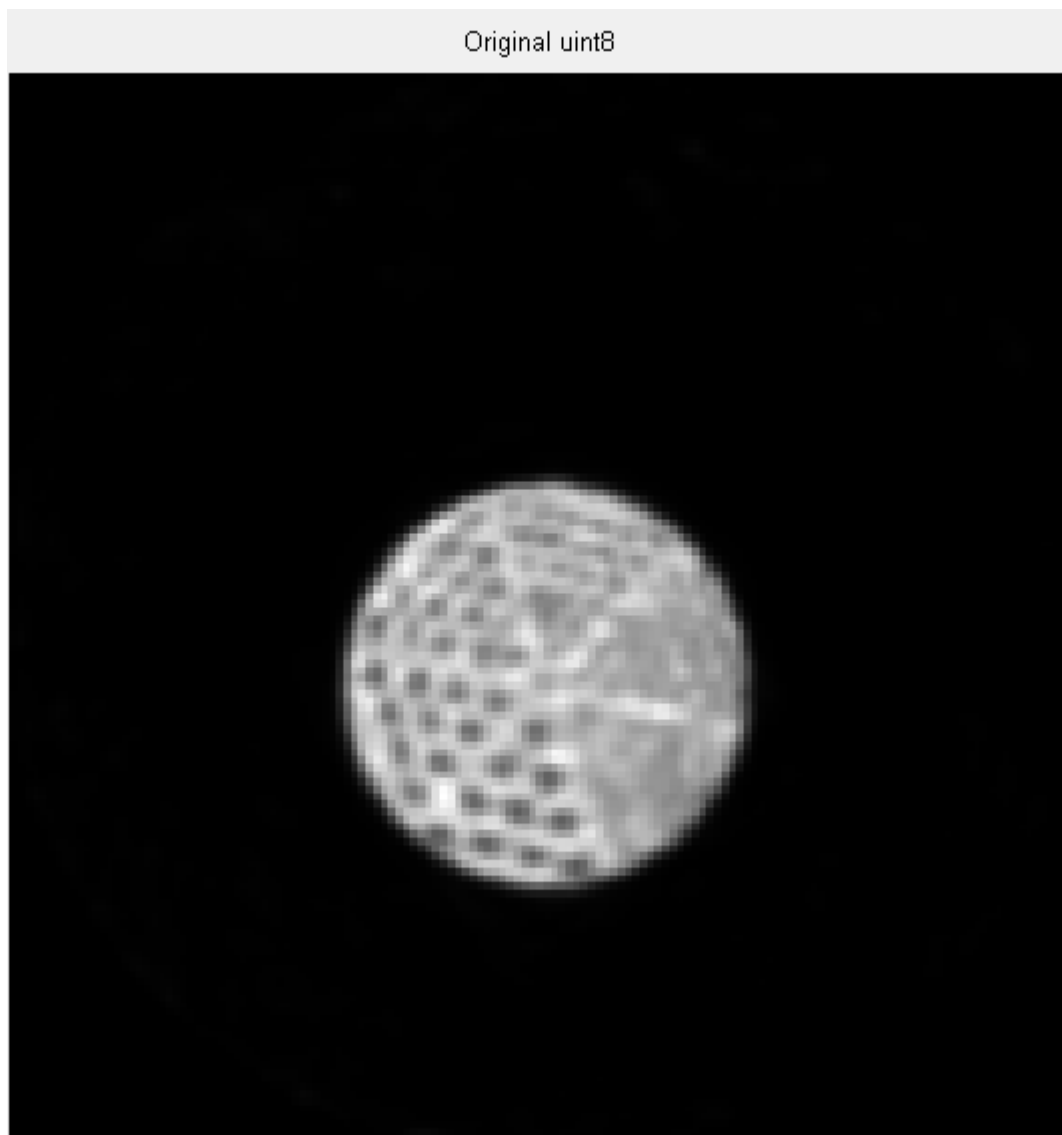


Figura 17: Plano original nº45 em uint8

O tempo de segmentar o plano é nos dados por uma barra de progresso, neste caso a duração é cerca de um minuto e meio, tudo depende do número de objetos que cada plano contém assim o número de cálculos que têm de ser efetuados em cada plano.

Nas duas figuras nº 18 vamos visualizar o valor médio dos objetos na gama UINT8, nº 19 o melhor contraste encontrado no melhor limiar, tudo na gama UINT8.

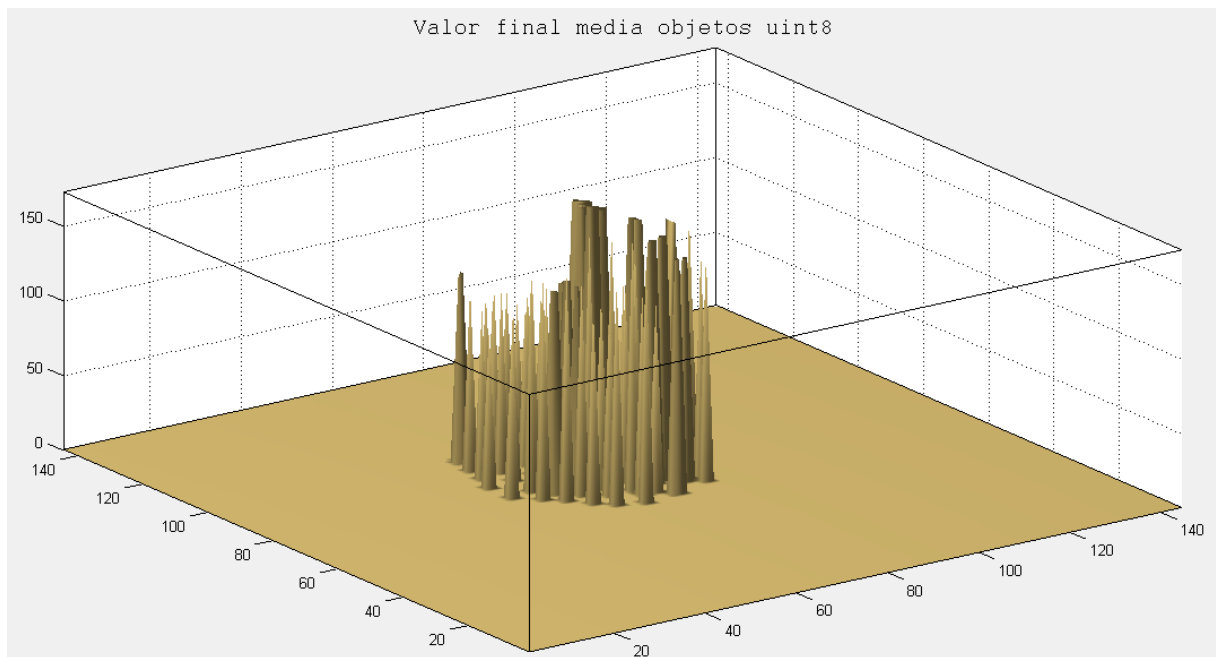


Figura 18: Valor médio dos objetos diferenciados do plano nº45 em uint8

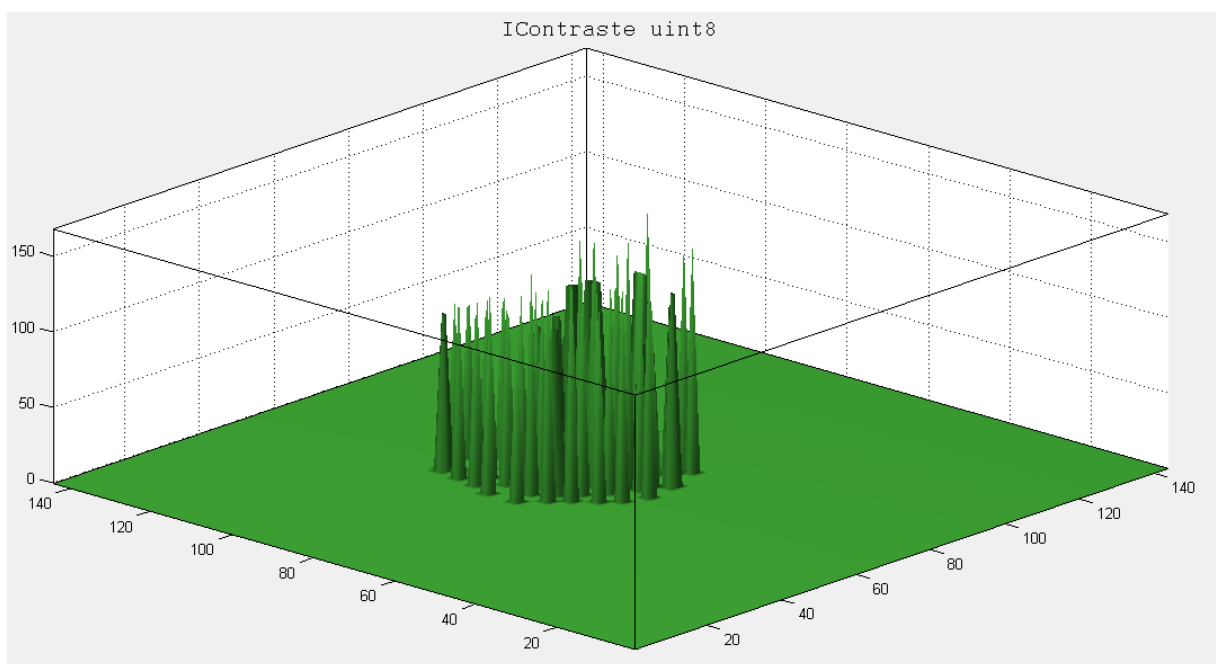


Figura 19: Valor do melhor contraste no limiar do plano nº45 em uint8

Podemos visualizar em 2D para este mesmo plano a localização do número de objetos assim como a sua área e a identificação do número do objeto na figura nº 20, também podemos ter acesso a uma listagem, no quadro nº 1, com os valores antes referidos, valor médio dos objetos e o melhor contraste tanto para UINT8 como UINT16 (2013, Regiões).

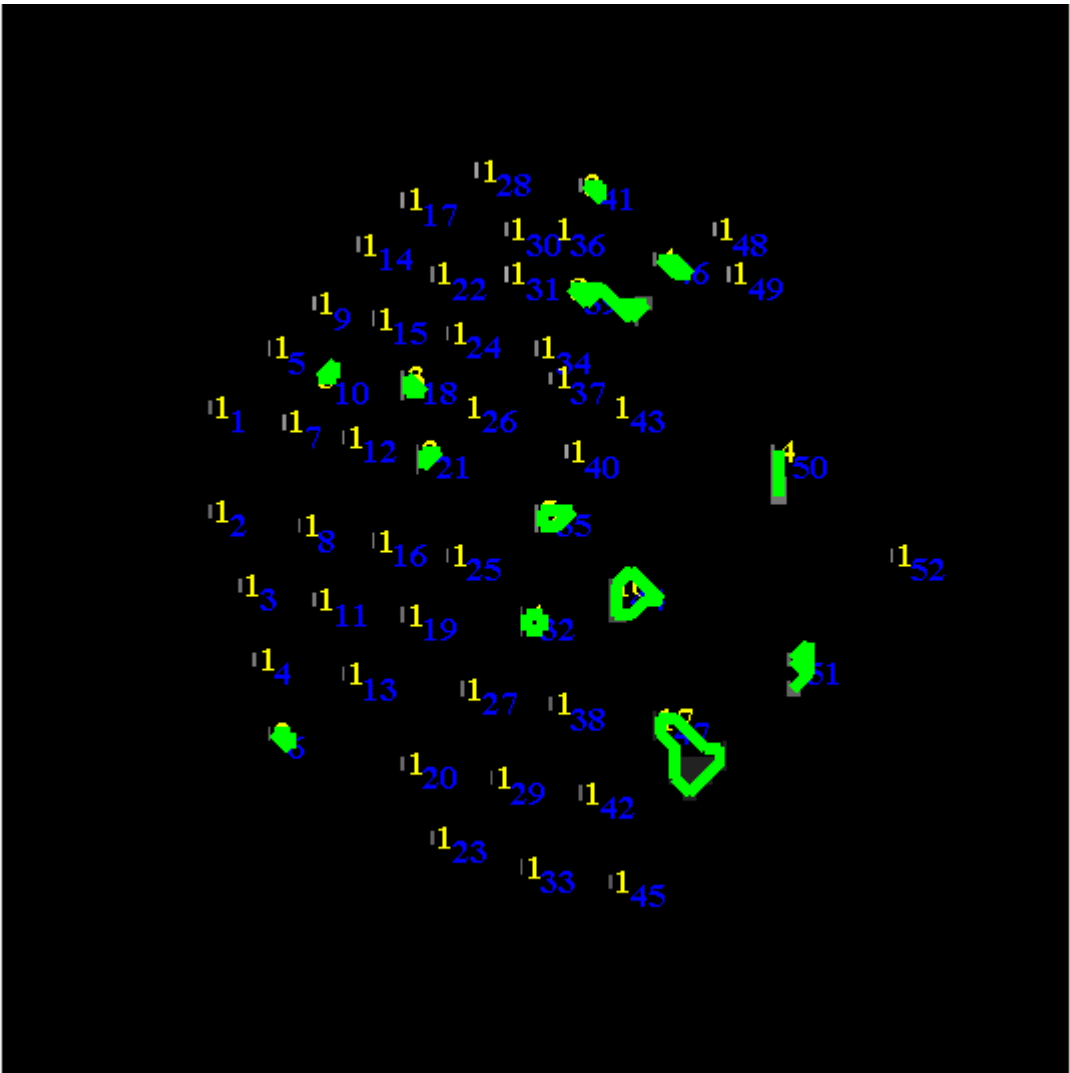


Figura 20: Identificação de localização dos objetos no plano nº45 em uint8

	Nº Objecto	Área do Objecto	Valor Médio uint8	Constraste uint8	Valor Médio uint16	Cor
1	1	1	107	148	9691	
2	2	1	105	150	9510	
3	3	1	114	141	10325	
4	4	1	122	133	11049	
5	5	1	146	109	13223	
6	6	3	126	152	11412	
7	7	1	147	108	13314	
8	8	1	116	139	10506	
9	9	1	152	103	13766	
10	10	3	135	146	12227	
11	11	1	123	132	11140	
12	12	1	132	123	11955	
13	13	1	114	141	10325	
14	14	1	129	126	11683	
15	15	1	142	113	12861	
16	16	1	122	133	11049	
17	17	1	147	108	13314	
18	18	3	133	146	12046	
19	19	1	109	146	9872	
20	20	1	104	151	9419	
21	21	3	132	147	11955	

Quadro 1: Listagem de valores dos objetos do plano nº45 em uint8 e uint16

Para finalizar este capítulo de fantasmas de PET vamos só visualizar a intensidade do melhor contraste e do valor médio dos objetos em 2D, segue nas figuras nº 21 e nº 22.



Figura 21: Intensidade do melhor contraste dos objetos do plano nº45 em uint8

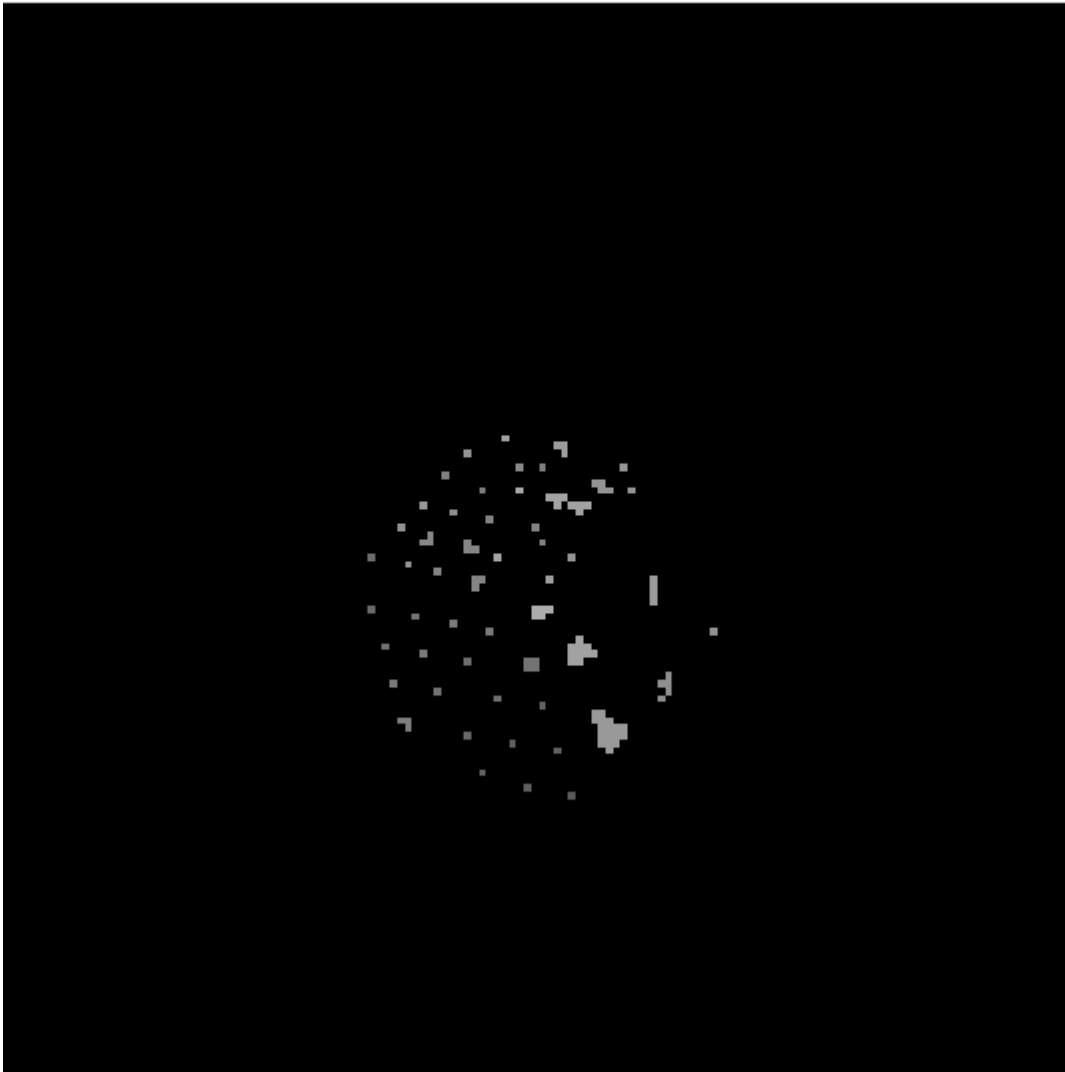


Figura 22: Intensidade do valor médio dos objetos do plano nº45 em uint8

Capítulo 5 – Segmentação imagem (humana) PET em 2D

Inicialmente começamos por mostrar a imagem original em uint8 já convertida de int16, para ser tratada para encontrar os objetos, ver a figura nº 23.

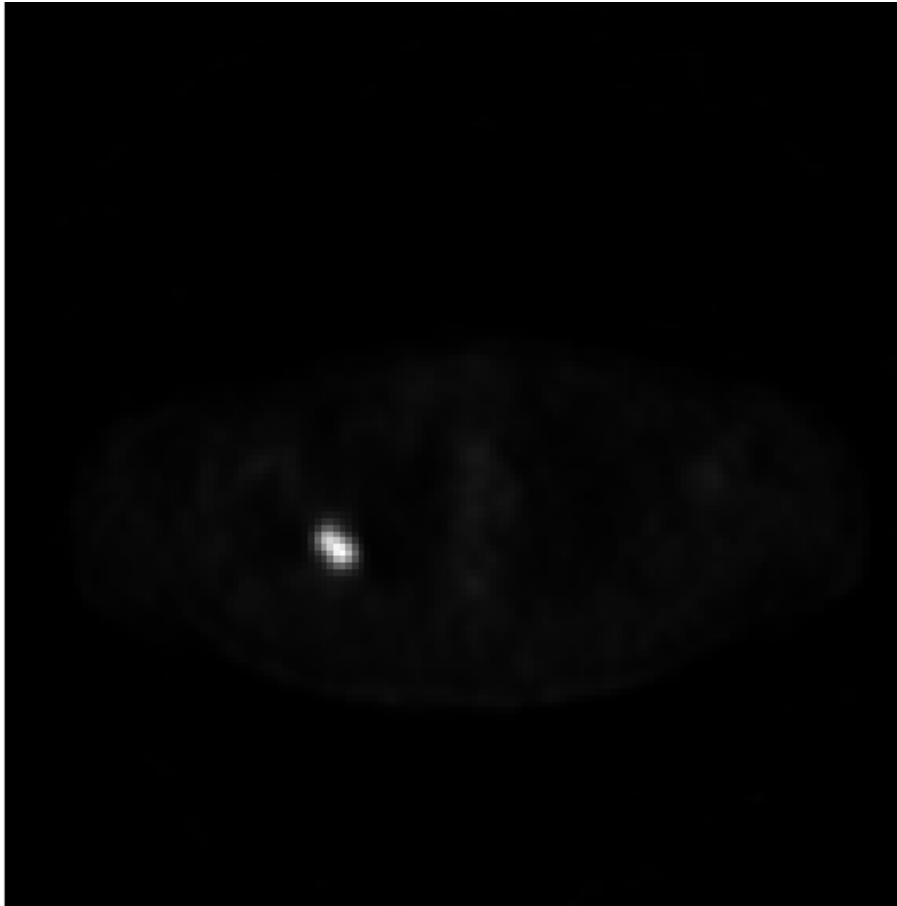


Figura 23: Plano original uint8 nº 174

A figura nº 24 mostra a representação tridimensional do plano nº 174 na gama de valores UINT8.

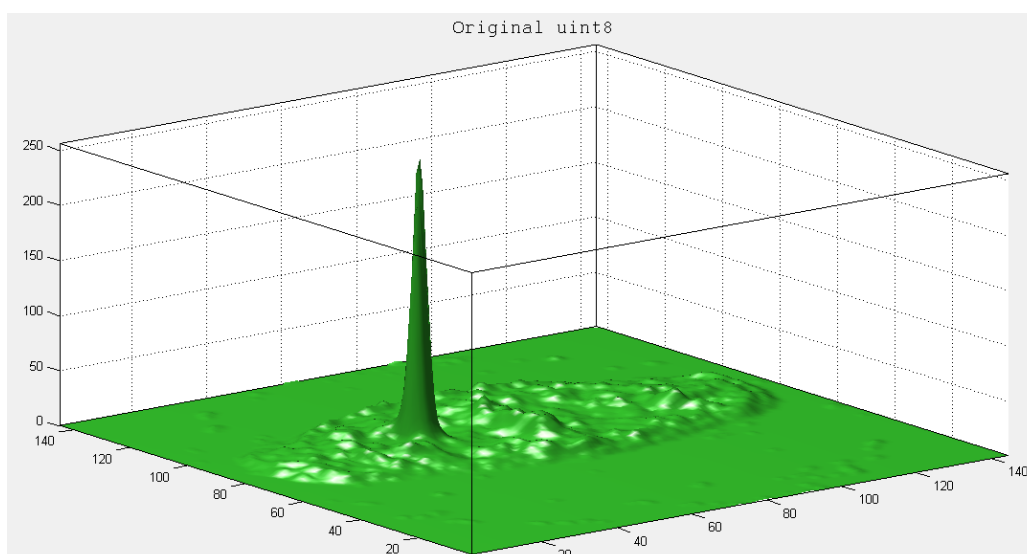


Figura 24: Representação tridimensional do plano nº174 em uint8

Dado que foi alterada a gama de valores para UINT8, vamos mostrar na figura nº 25 a representação tridimensional mas na gama de valores INT16 da imagem original.

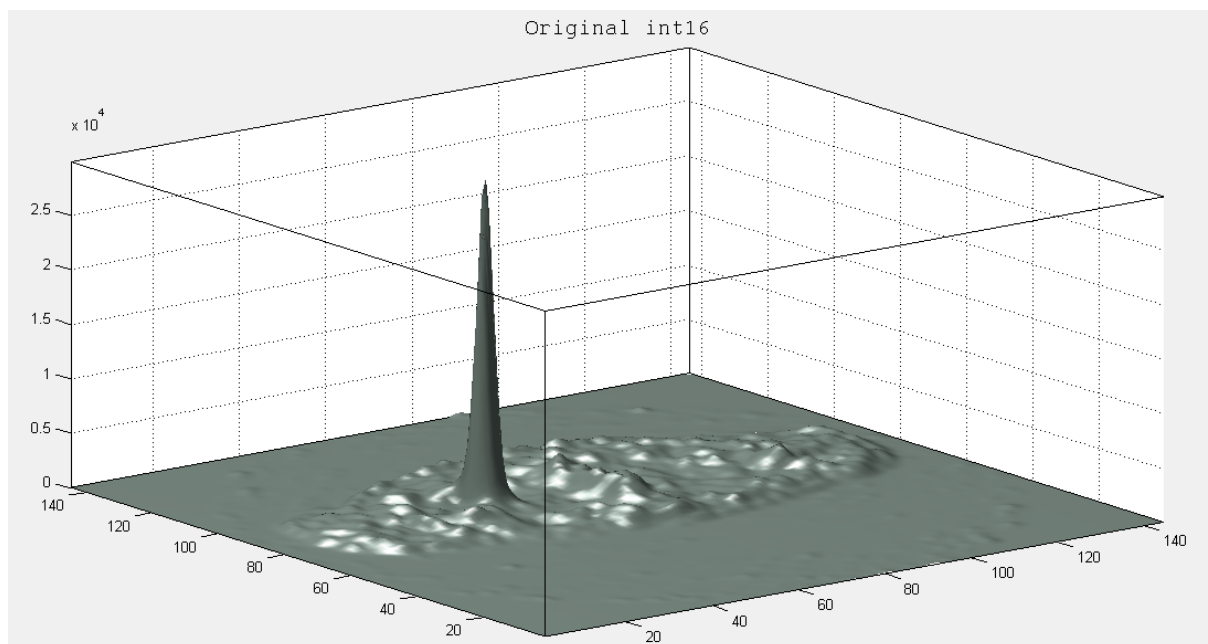


Figura 25: Representação tridimensional do plano nº 174 original int16

A segmentação deste plano demora cerca de 10 a 20 segundos, neste caso visto não ser um plano muito complexo, também tudo depende do número de objetos que o algoritmo deteta no plano, o decorrer da segmentação é dado através de uma barra de progresso do seguinte estilo abaixo (2012, *Matlab Progressbar*).

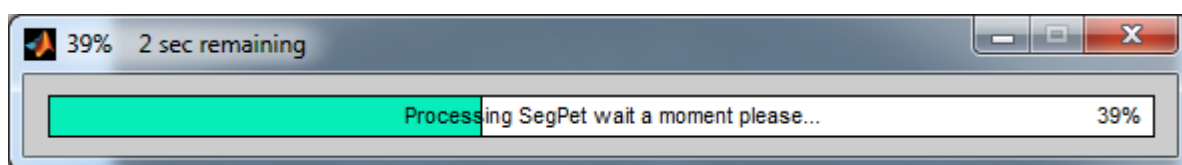


Figura 26: Barra de progresso do plano

Na imagem original UINT8 são encontrados os seguintes objetos, com o seguinte nível de intensidade, ver figura nº 27.

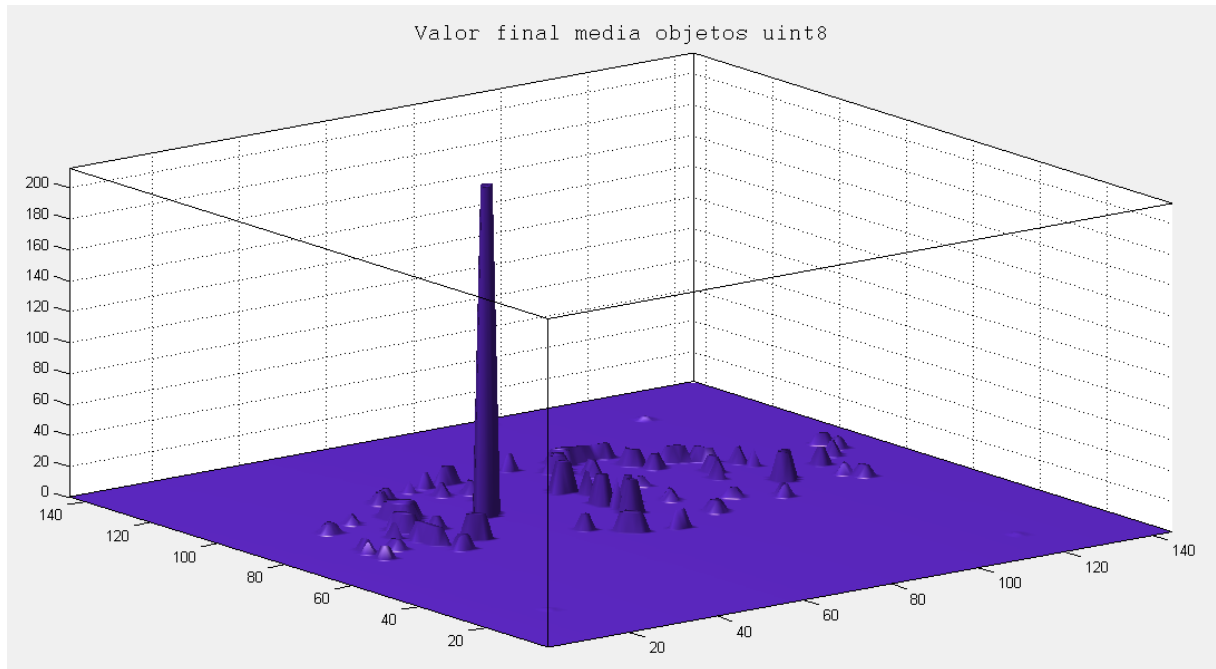


Figura 27: Representação tridimensional dos objetos encontrados no plano nº 174 em uint8

Na figura nº 27 os níveis de intensidade dos objetos são calculados a partir da imagem original uint8, para calcular o contraste inverteu-se a imagem original uint8 ou seja o que era branco passa a preto e o que é preto passa a branco.

A fórmula (8) para inverter não é mais do que uma subtração ou seja a nova matriz (In_double_uint8) igual a 255 menos matriz original (Original_double_uint8).

$$\text{Matriz Inversa} = 255 - \text{Matriz Original Double} \quad (8)$$

Ao fazer isto teremos uma representação tridimensional mais fácil de entender, porque basta olhar para figura nº28 veremos que mesmo só pelas coordenadas sabemos qual o objeto que tem contraste mais elevado.

Caso se inverta a matriz para calcular a média de cada objeto, a representação tridimensional iria ser o inverso do que vemos na figura nº 27 porque o objeto de maior contraste teria valores perto de zero.

A figura nº 28 mostra a representação tridimensional do melhor contraste encontrado neste plano para cada objeto no limiar de 0 até 255.

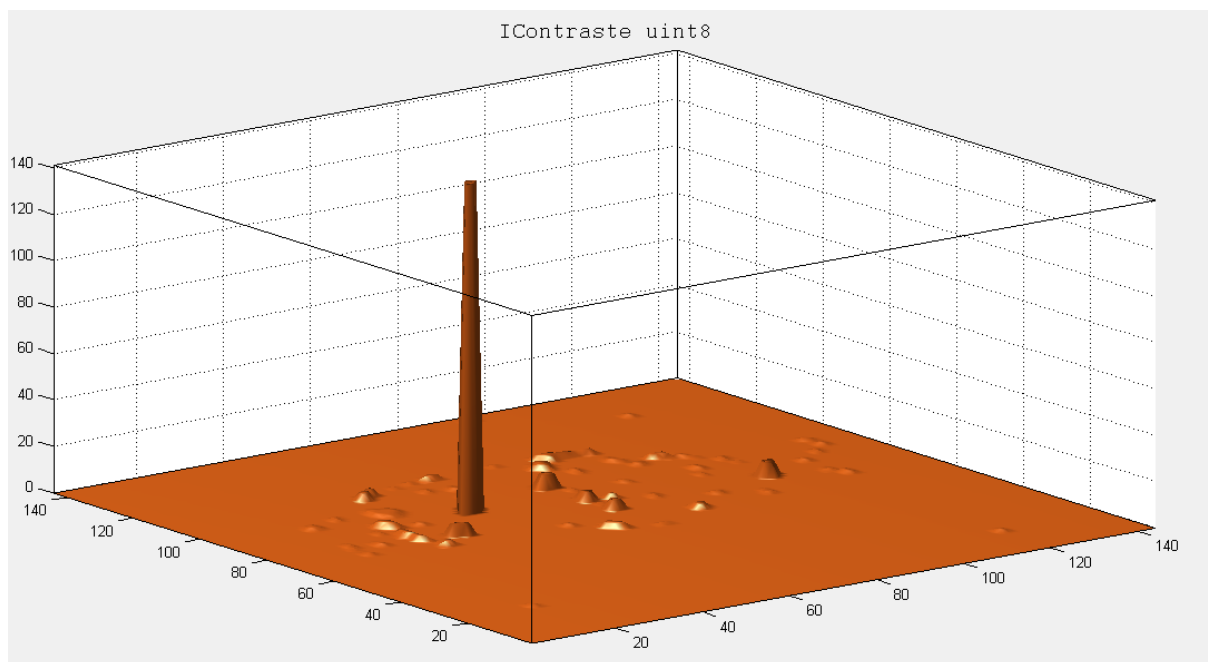


Figura 28: Representação tridimensional do melhor contraste uint8 no limiar

O quadro nº 2 mostra o número de objeto, a sua área, valor médio do objeto em UINT8 e UINT16, contraste do objeto em UINT8 e UINT16, como também podemos saber o número total de objetos encontrados no plano segmentado, esta tabela pode ser exportada para *excel* (2012, *Exporting to Excel*).

	Área do Objecto	Valor Médio uint8	Constraste uint8	Valor Médio uint16	Constraste uint16
1	3	1	1	117	117
2	3	8	1	933	117
3	4	8	1	933	117
4	8	6	1	700	117
5	5	6	1	700	117
6	2	10	1	1167	117
7	3	6	1	700	117
8	17	11	2	1283	233
9	14	11	2	1283	233
10	6	9	2	1050	233
11	32	10	1	1167	117
12	9	16	5	1867	583
13	4	8	1	933	117
14	11	7	3	817	350
15	11	212	141	24735	16451
16	1	8	1	933	117
17	3	8	1	933	117
18	2	10	1	1167	117
19	4	11	1	1283	117
20	4	10	1	1167	117
21	5	8	1	933	117

Quadro 2: Valores do objeto de elevado contraste no plano 174 humano

Podemos verificar que o objeto número 15, é detentor do mais elevado contraste, com uma área igual a 11 *pixels*, uma média do objeto de 212.

Podemos ver a identificação de todos os objetos no plano que estamos a segmentar, figura abaixo mostra número, área e contorno do objeto de maior elevado contraste.

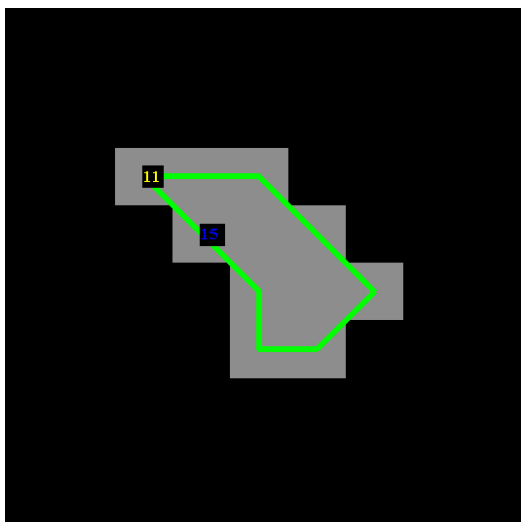


Figura 29: Area, número e contorno de objeto

A imagem abaixo mostra nos que este é o único objeto com significado de interesse neste plano nº174 de volume de imagem humana.

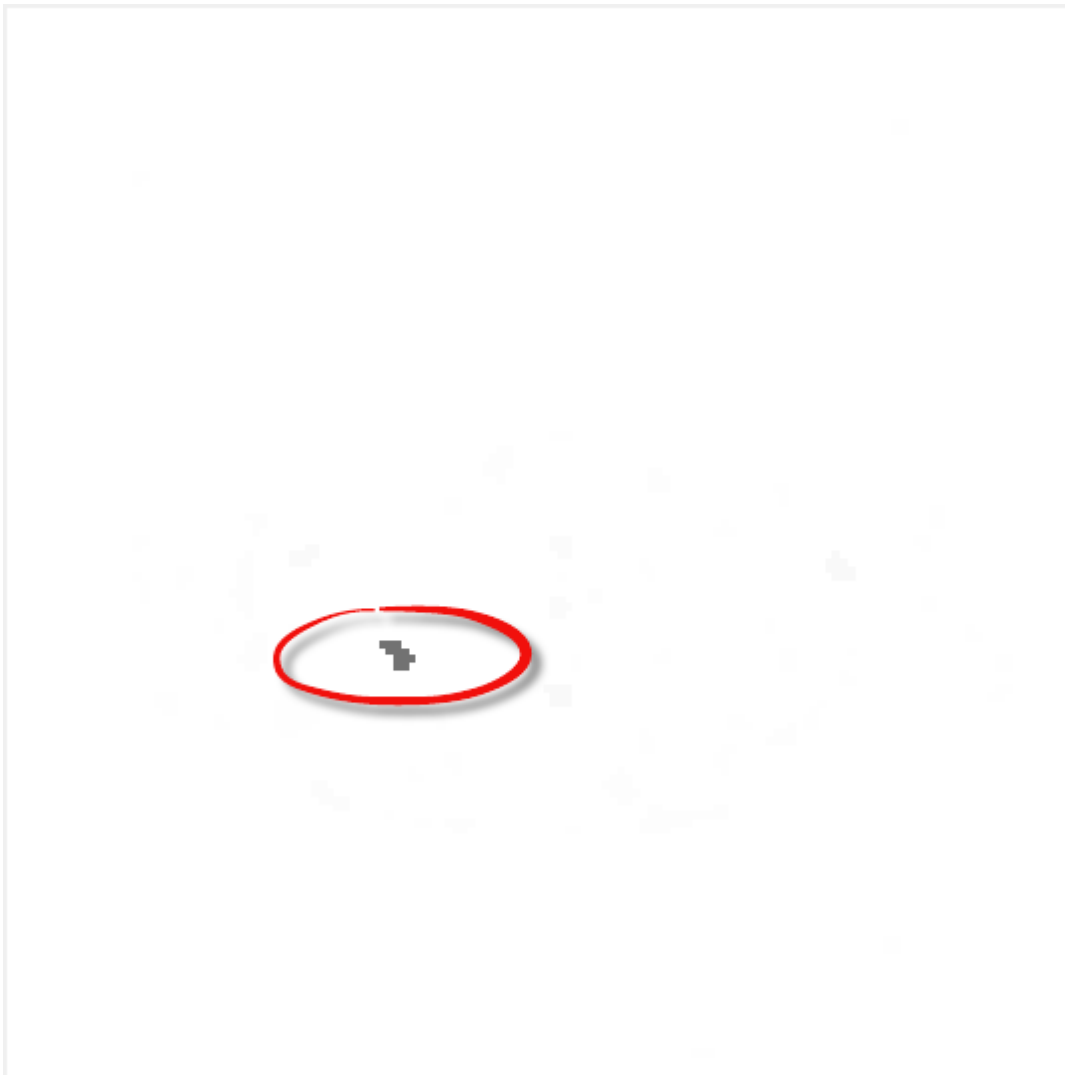


Figura 30: Objeto de elevado contraste plano nº174

Por fim podemos visualizar na figura nº 31, a forma e a intensidade do valor médio dos objetos encontrados no plano segmentado, representado na figura abaixo.

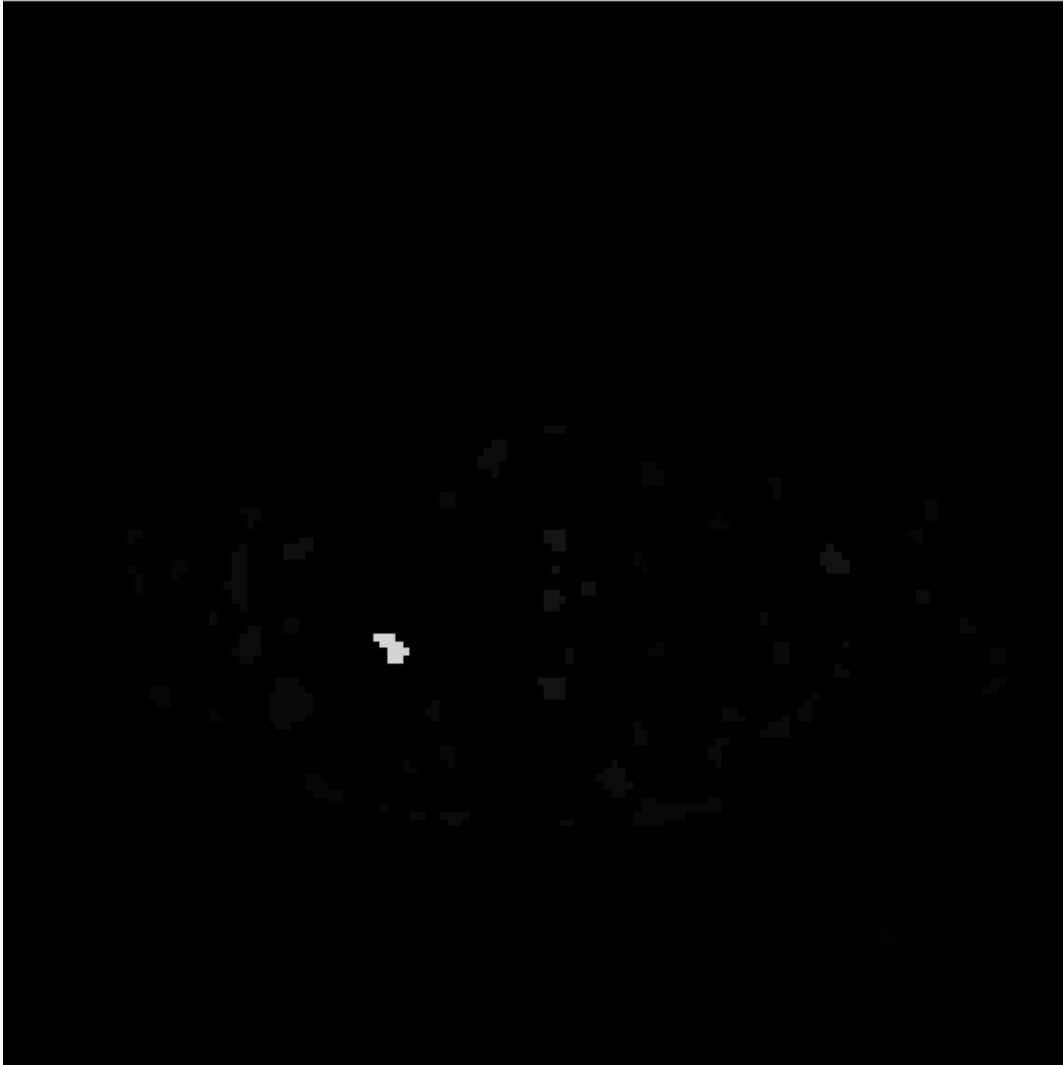


Figura 31: Forma e intensidade dos objetos segmentados em uint8

Capítulo 6 – Construção de Volume imagens (fantomas) PET 3D

Após segmentar algumas imagens de fantasmas, não tinha a noção do que se tratava a nível de volume de imagens, por isso procedi à construção do volume de imagens para ter a perceção do que realmente eu estaria a segmentar (2013, *Dicom Read Volume*).

A figura nº 32 mostra o volume de fantasmas na gama de valores UINT8, visto se construísse em INT16 ou UINT16 iria se tornar pesado a imagem visto que a gama de valores é muito alta (2013, *Uitab customizations*).

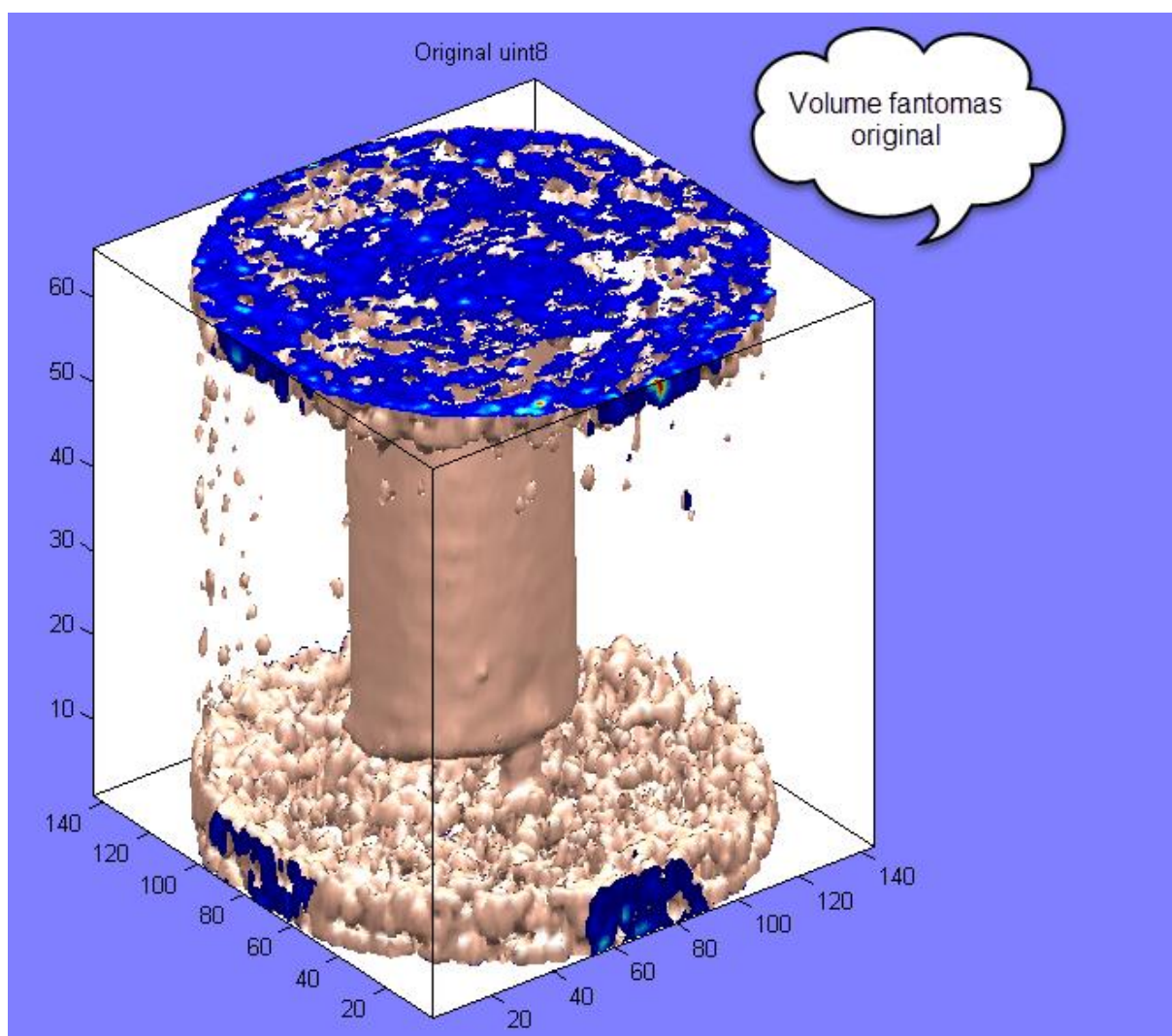


Figura 32: Volume fantoma inicial em uint8

Após carregar o volume original (144;144;66) podemos ter acesso às seguintes funcionalidades (As funcionalidades estão explicadas na segmentação de imagem humana 3D, com uma única diferença, o numero de planos, sendo neste caso 66);

- Carregar matriz já segmentada;
- Apagar matriz apresentada;
- Segmentar volume;
- Corte axial;
- Corte coronal;
- Corte sagital;
- Permite ainda os 3 cortes em simultâneo;
- Rotação de volume;
- Aumentar imagem;
- Reduzir imagem;

Podemos observar a figura nº 33 com um corte axial no plano nº 45 do volume original de fantasmas.

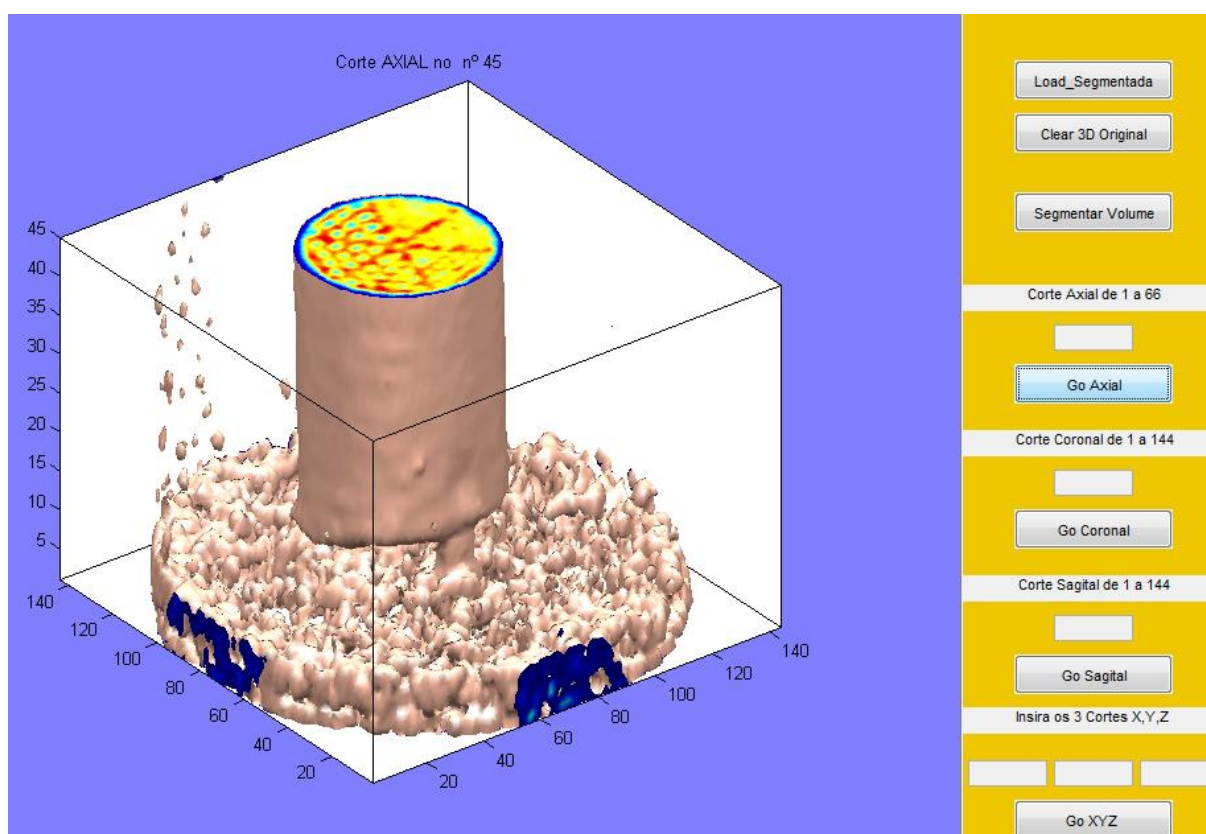


Figura 33: Corte axial nº 45 volume original fantasmas

O tempo de segmentação de um volume de fantasmas é necessariamente cerca de 60 minutos, para ter controlo de tempo de plano a ser segmentado e estado do volume segmentado introduzimos duas barras de estado de progresso com tempo.

Isto facilita o controlo da segmentação para saber o ponto da situação num dado instante durante o tempo de segmentação do volume.

A figura abaixo mostra nos a duas barras, com progresso do volume e do plano.

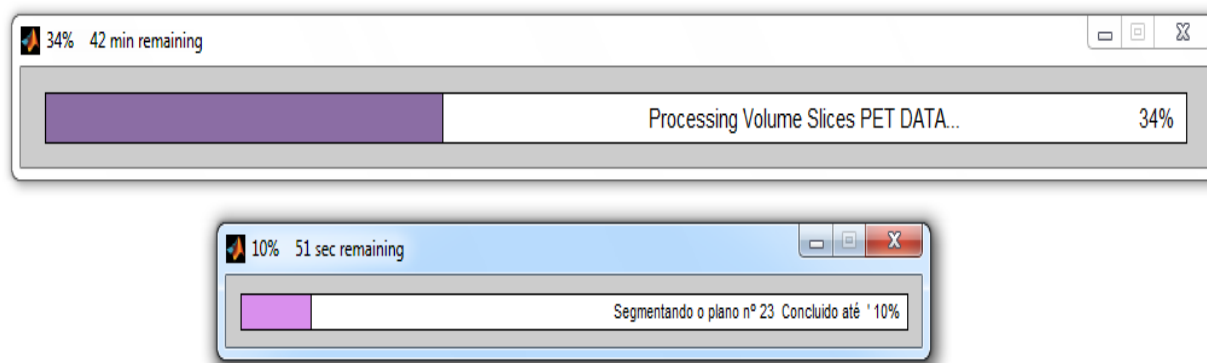


Figura 34: Barras de progresso

Após concluída a segmentação do volume de fantasmas podemos visualizar todo volume segmentado numa matriz 3D, dado que este só tem 66 planos logo a matriz será (144;144;66).

Dada a conclusão da segmentação do volume de imagens fantoma obtemos o seguinte, ver figura nº 35;

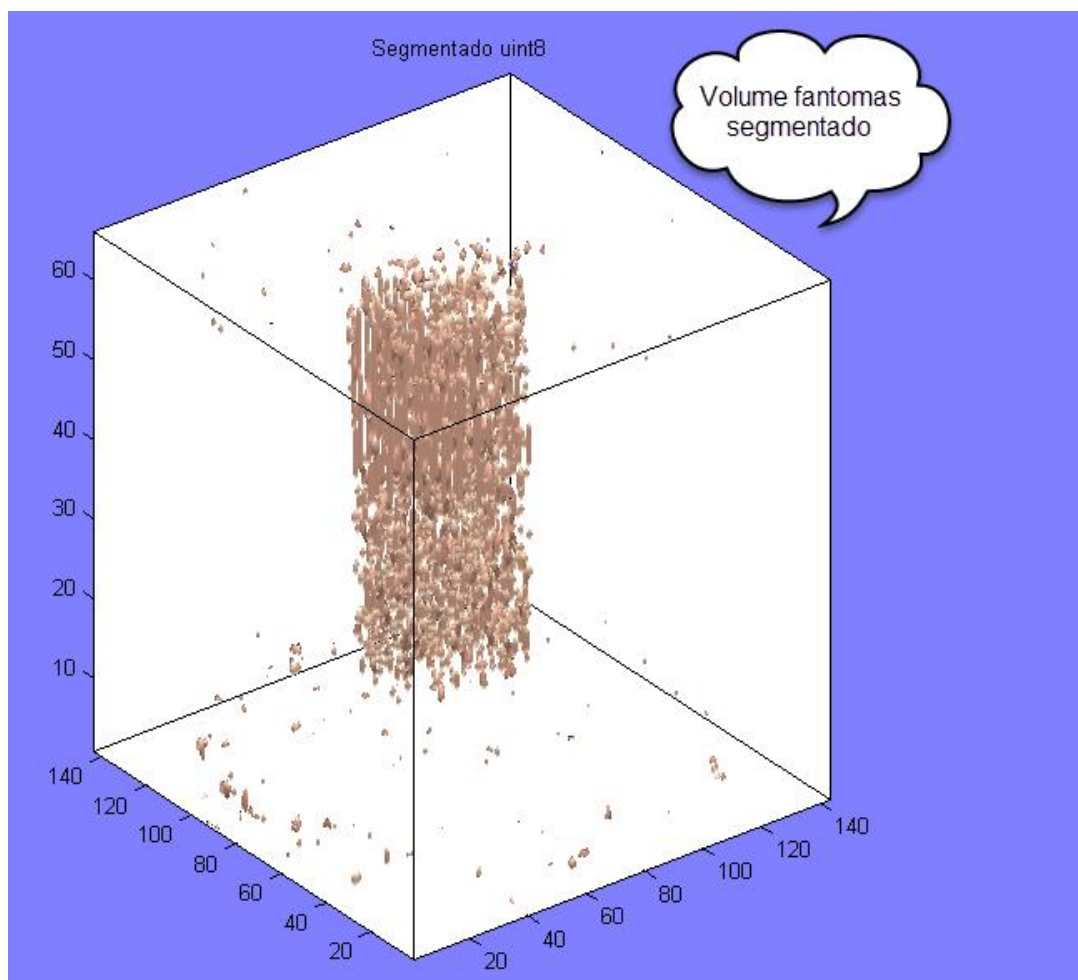


Figura 35: Volume fantomas segmentado uint8

No separador onde visualizamos o volume de fantasmas segmentado teremos as seguintes funcionalidades (As funcionalidades encontram explicadas na segmentação de imagem humana 3D, com uma única diferença, sendo a altura do volume, neste caso 66 planos);

- Criar vídeos;
- Apagar volume segmentado;
- Corte axial;
- Corte coronal;
- Corte sagital;
- Permite ainda os 3 cortes em simultâneo;

Na figura nº 36 podemos observar o mesmo corte que fizemos no volume original de fantasmas, onde podemos ver as regiões diferenciadas encontradas.

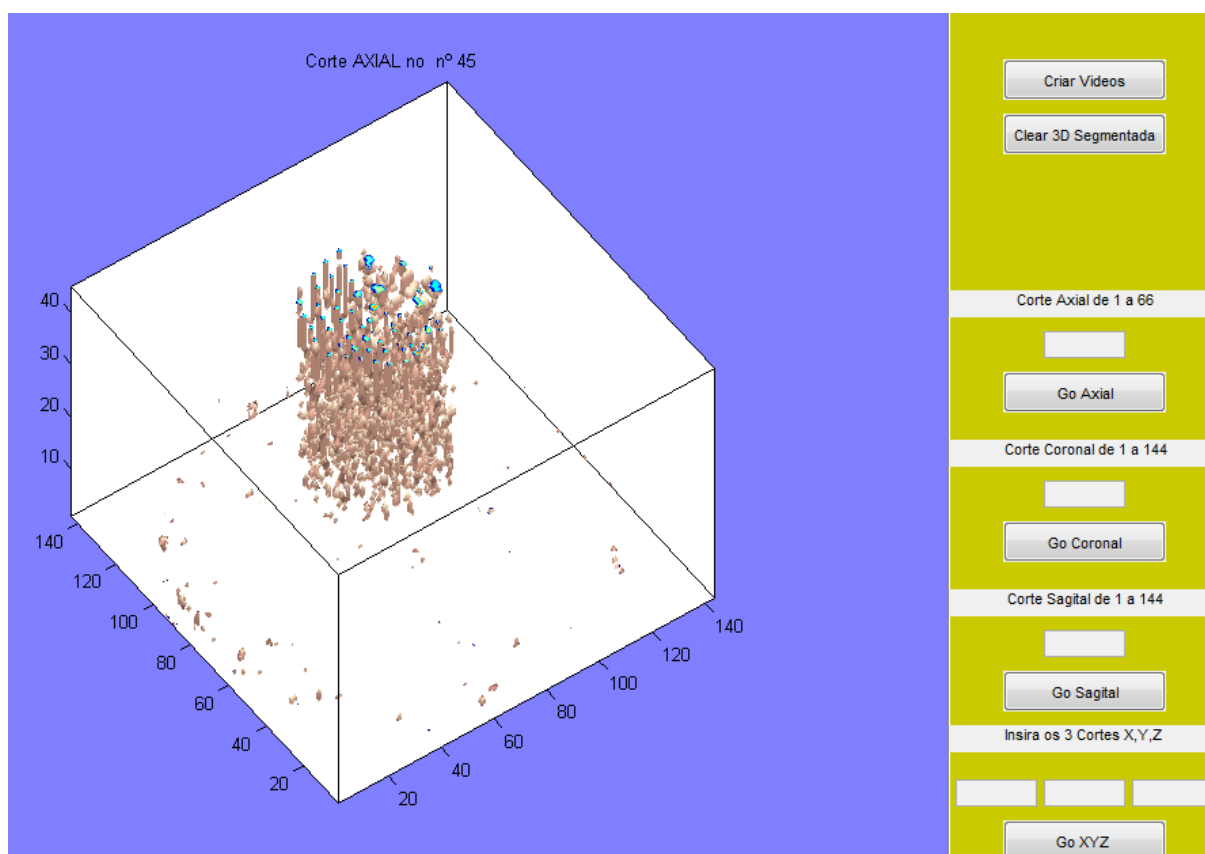


Figura 36: Corte axial nº45 volume de fantasmas segmentado

Ao clicar em criar vídeos iremos ter um novo separador onde podemos visualizar em vídeo o volume de fantasmas segmentado de três formas diferentes.

Primeiro vídeo mostra a intensidade do contraste de cada plano do volume, segundo exclui do volume todos os objetos de baixa intensidade até um dado limiar parametrizado e por fim o terceiro vídeo é uma visualização 2D de todos os planos onde os objetos de baixa intensidade adquirem a cor azul, os de média a cor verde e os de alta a cor encarnada.

As três figuras nº 37, nº 38 e nº 39 serão as formas de visualização no modo pausado dos vídeos gerados.

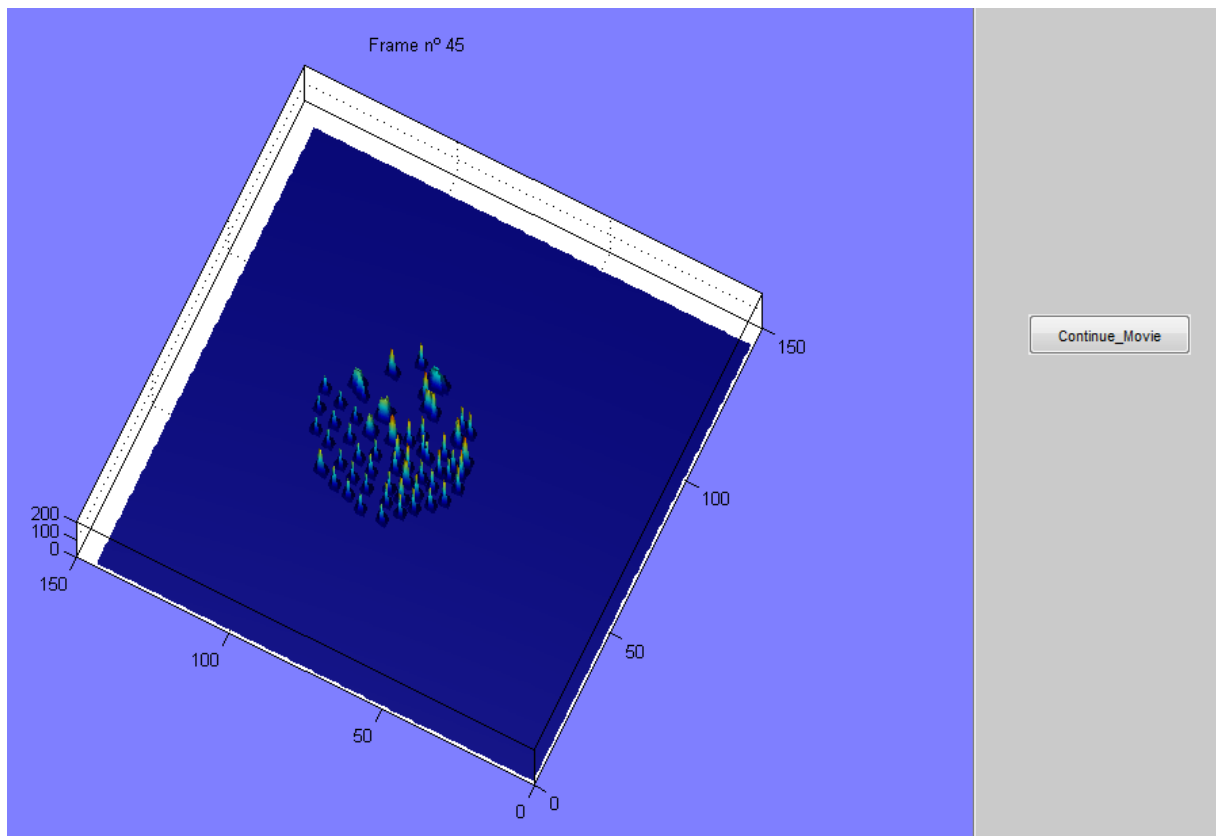


Figura 37: Objetos fantasmas pausa do plano nº45

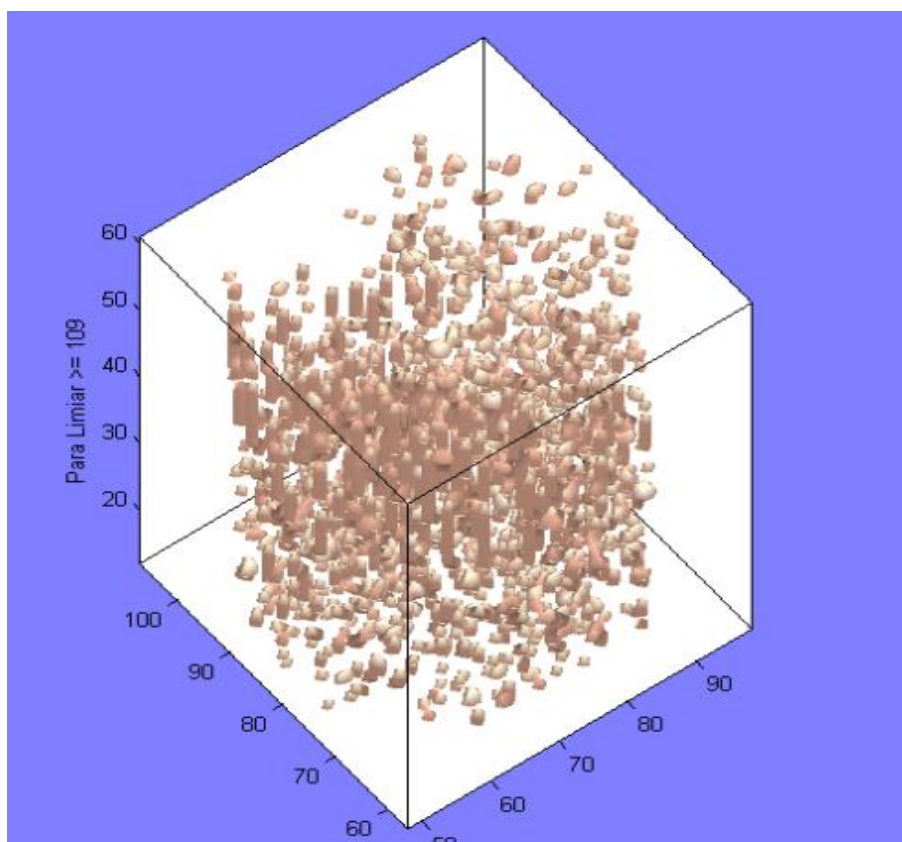


Figura 38: Exclusão de objetos fantasmas pausa limiar 109

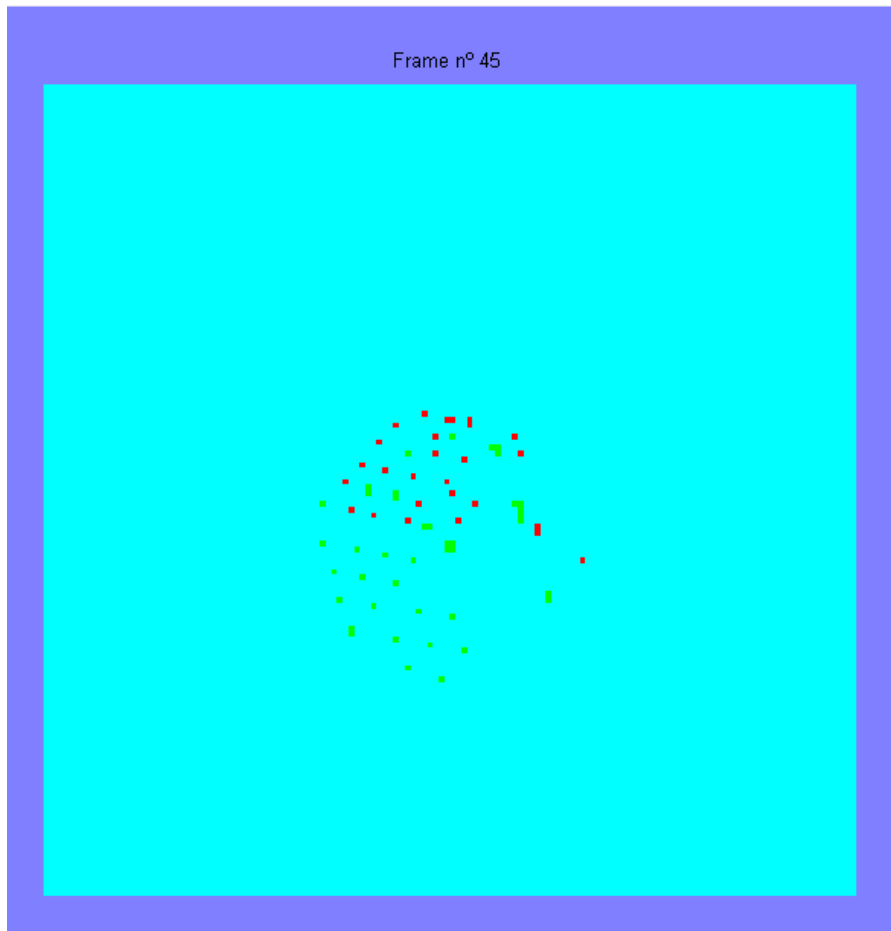


Figura 39: Plano fantoma 45 em pausa

Capítulo 7 – Construção de Volume imagens (humana) PET em 3D

Para se perceber melhor como funciona tudo o que inicialmente fizemos nos capítulos anteriores, vamos mostrar os planos todos juntos ou seja o volume antes de se iniciar a segmentação de cada plano. Como podemos verificar na figura n° 40, temos 255 planos, onde cada plano assenta sobre uma matriz de 144 linhas por 144 colunas (2013, *Techniques for Visualizing Scalar Volume Data*).

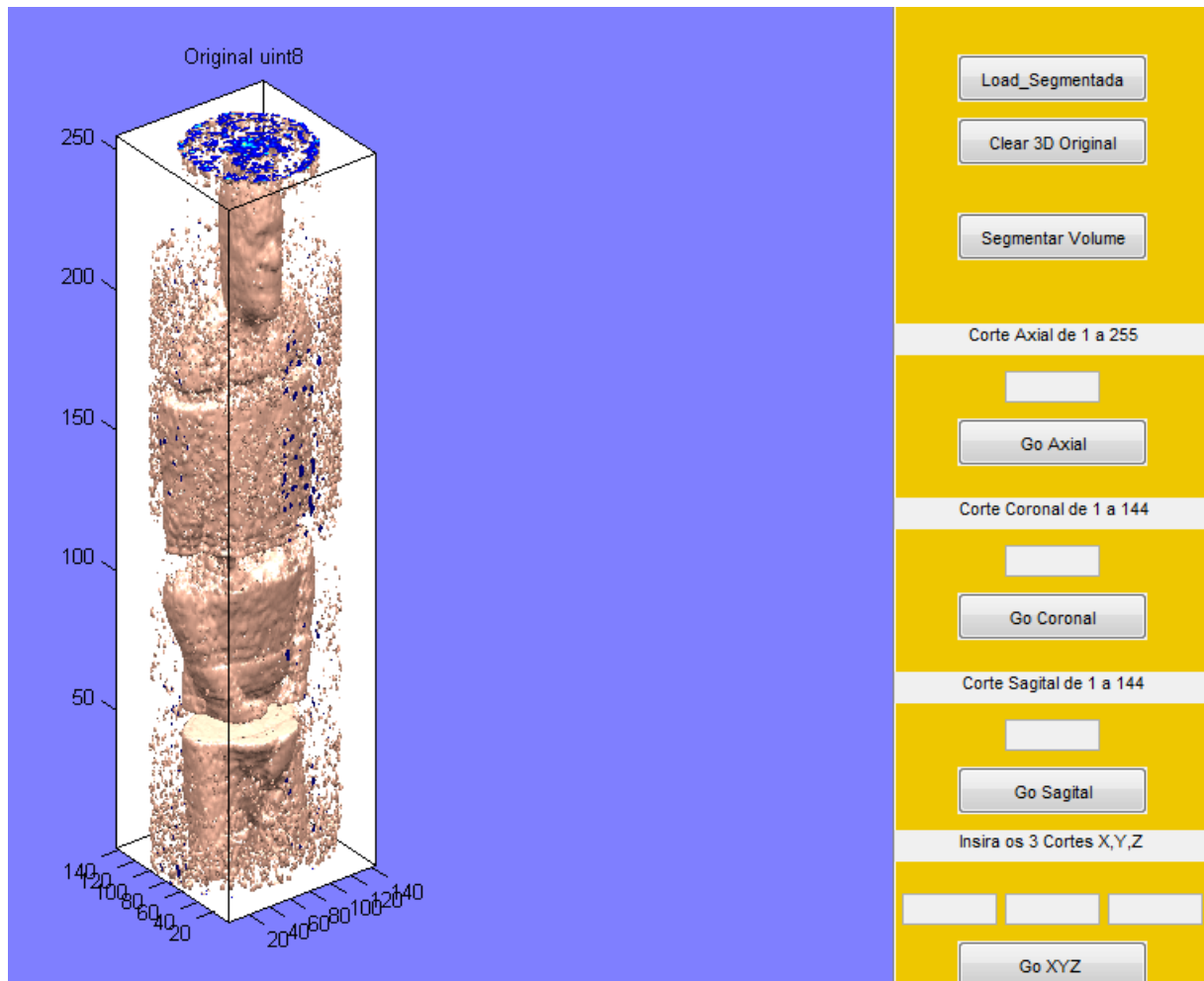


Figura 40: Volume inicial uint8

Após carregar este volume de imagens para visualizar na matriz 3D como (144;144;255), temos as seguintes funcionalidades;

- Carregar matriz já segmentada, esta funcionalidade funciona unicamente no *Matlab*, porque não foi possível compilar executável com matrizes guardadas.
- Apagar matriz apresentada, necessariamente eliminar eixos de apresentação.

- Segmentar volume, rotina mais importante do programa (ver anexo).
- Corte axial é limitar a altura do volume entre 1 e 255.
- Corte coronal é limitar o número de linhas entre 1 e 144.
- Corte sagital é limitar o número de colunas entre 1 e 144.
- Permite ainda os 3 cortes em simultâneo é limitar os 3 cortes em simultâneo.
- Rotação de volume, propriedade do *Matlab*.
- Aumentar imagem, propriedade do *Matlab*.
- Reduzir imagem, propriedade do *Matlab*.

Na figura nº 41 podemos observar um corte axial na fatia nº 174 do volume original.

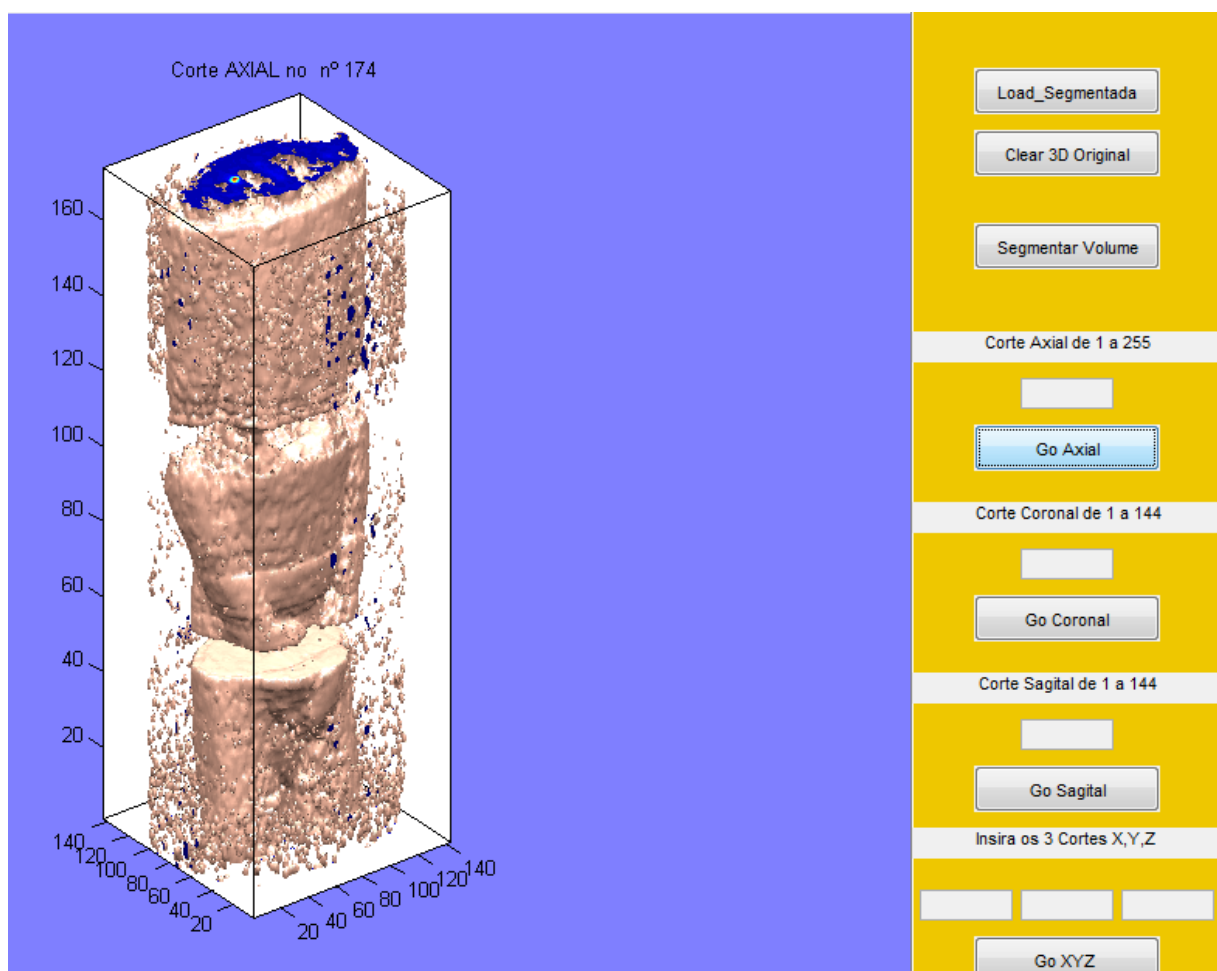


Figura 41: Corte axial nº 174 no volume original

Após a construção do volume inicial podemos segmentar plano a plano seguindo o método de uma segmentação bidimensional, começando em 1 até 255 guardando tudo em uma matriz tridimensional (144;144;255).

A segmentação do volume do corpo demora cerca de 45 minutos, o tempo estimado é dado através de uma barra de progresso, tanto a nível do volume total como de cada plano que está a ser segmentado (máquina utiliza um sistema de 32 *bits*).

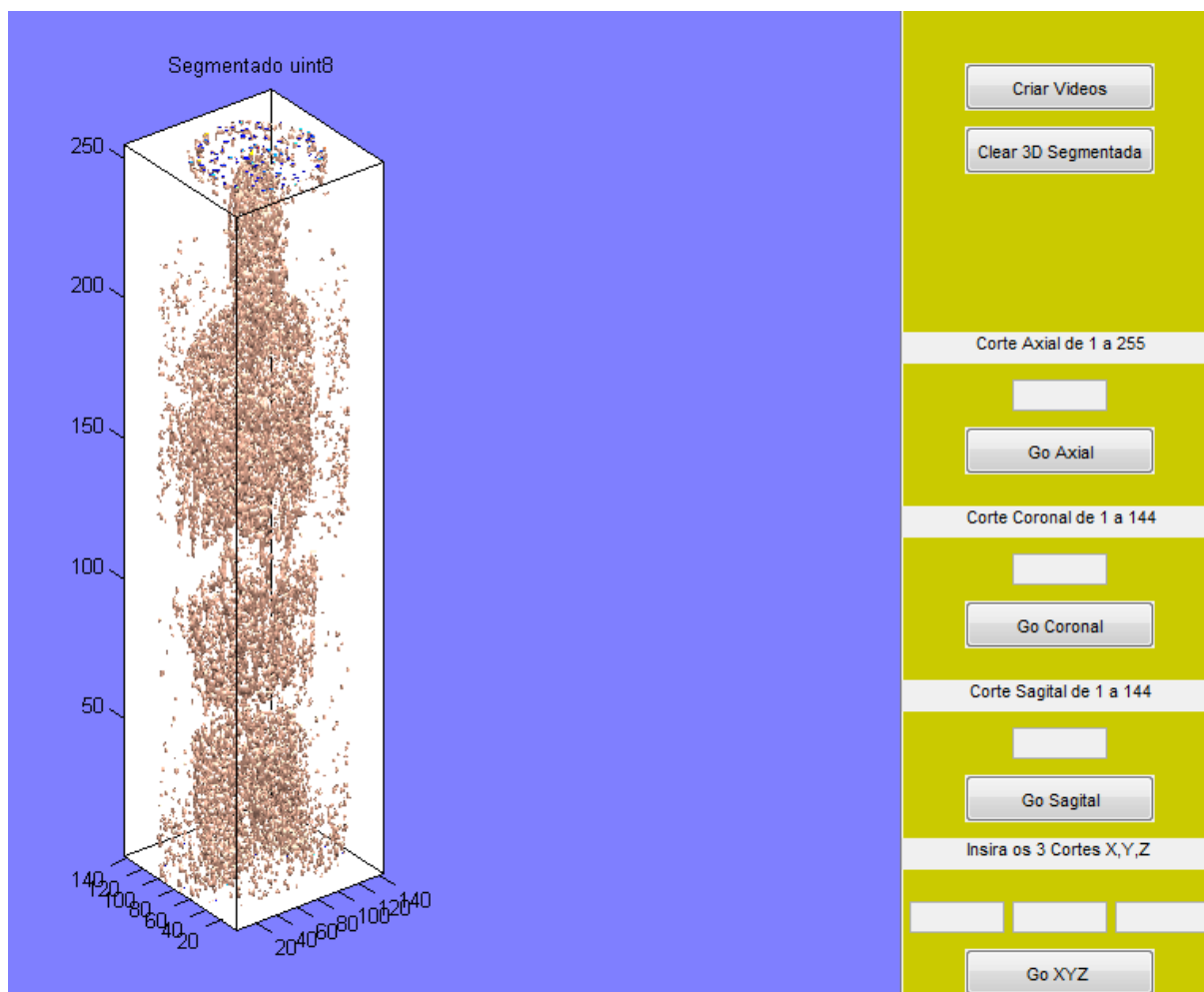


Figura 42: Volume de imagens segmentado em uint8

No separador que visualizamos o volume segmentado iremos ter as seguintes funcionalidades;

- A criação de vídeos é gerado a partir da matriz que visualizamos segmentada, sendo uma sequência de imagens geradas sobrepostas, com uma velocidade limitada para se tornar sequencial.
- Apagar volume segmentado, é necessariamente eliminar a matriz 3D.
- Corte axial é limitar a matriz (144;144;255) a um valor entre 1 e 255.
- Corte coronal é limitar o número de linhas entre 1 e 144.
- Corte sagital é limitar o número de colunas entre 1 e 144.

- Permite ainda os 3 cortes em simultâneo, ou seja limitar as 3 coordenadas da matriz em simultâneo.

Na figura abaixo podemos visualizar o corte axial nº 174, que mostra o objeto encontrado com um contraste elevado.

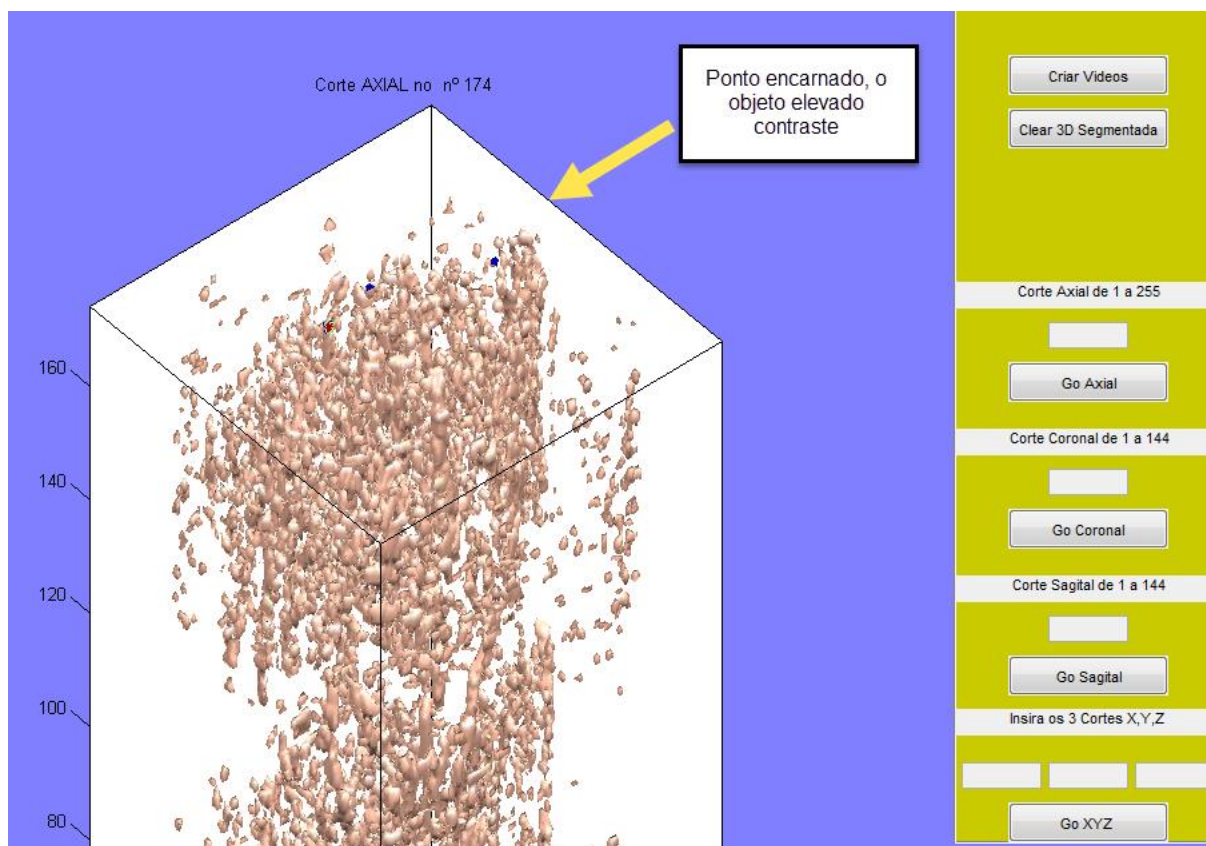


Figura 43: Corte axial nº174 no volume segmentado

Ao clicar na funcionalidade criar vídeos, iremos ter um novo separador, onde teremos a oportunidade de visualizar em vídeo o volume segmentado, com acesso a pausar o vídeo, estes são guardados na seguinte diretoria, “C:\Videos “.

As formas de vídeo que são criadas podem ser visualizadas nas seguintes figuras nº44, nº45, abaixo.

Primeira figura n º 44 mostra intensidade do contraste no plano n º 174.

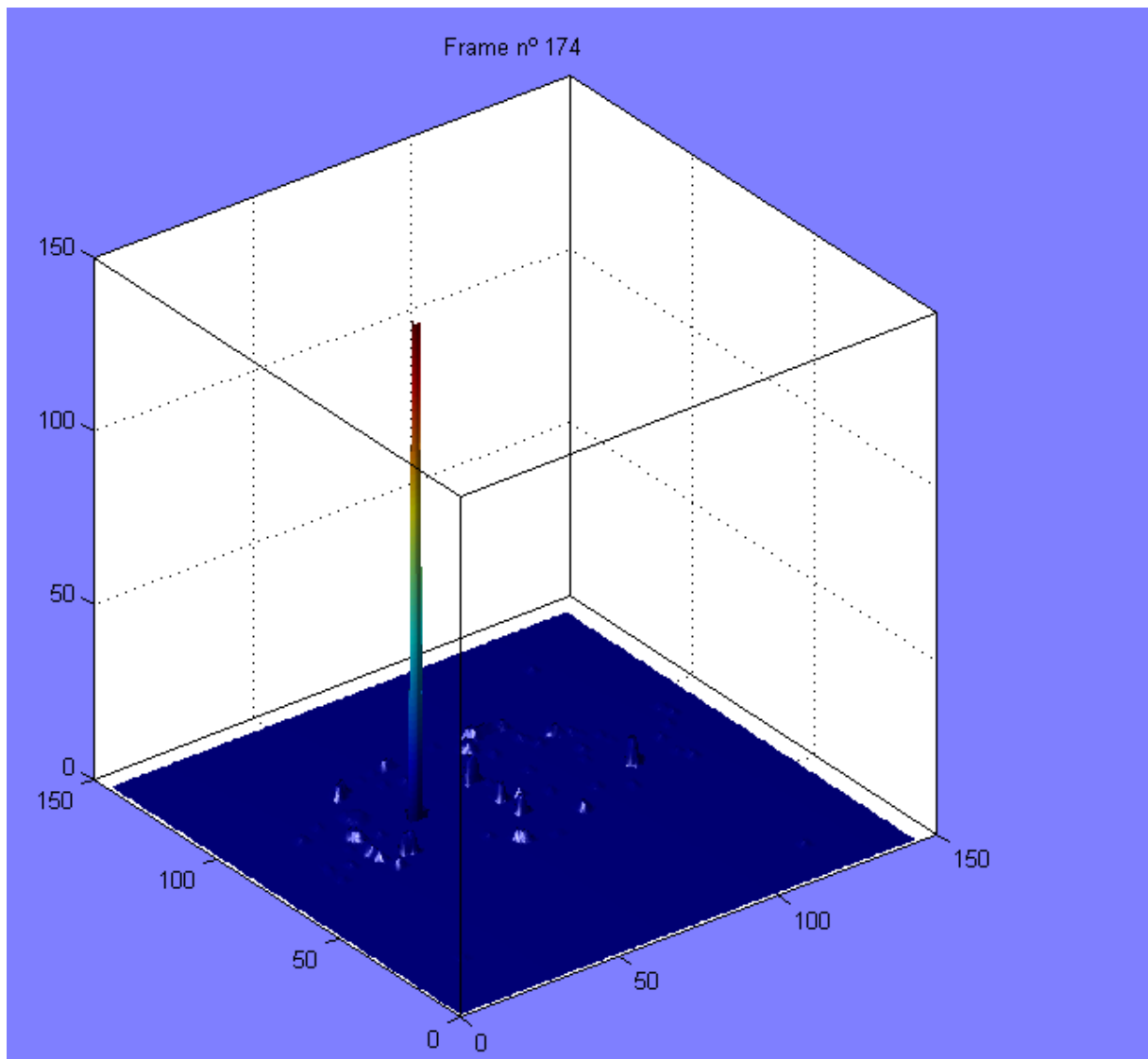


Figura 44: Pausa no plano nº 174

A figura nº 45 mostra os objetos com contraste abaixo do limiar excluídos, deixando visíveis somente todos aqueles acima do limiar em que se encontramos.

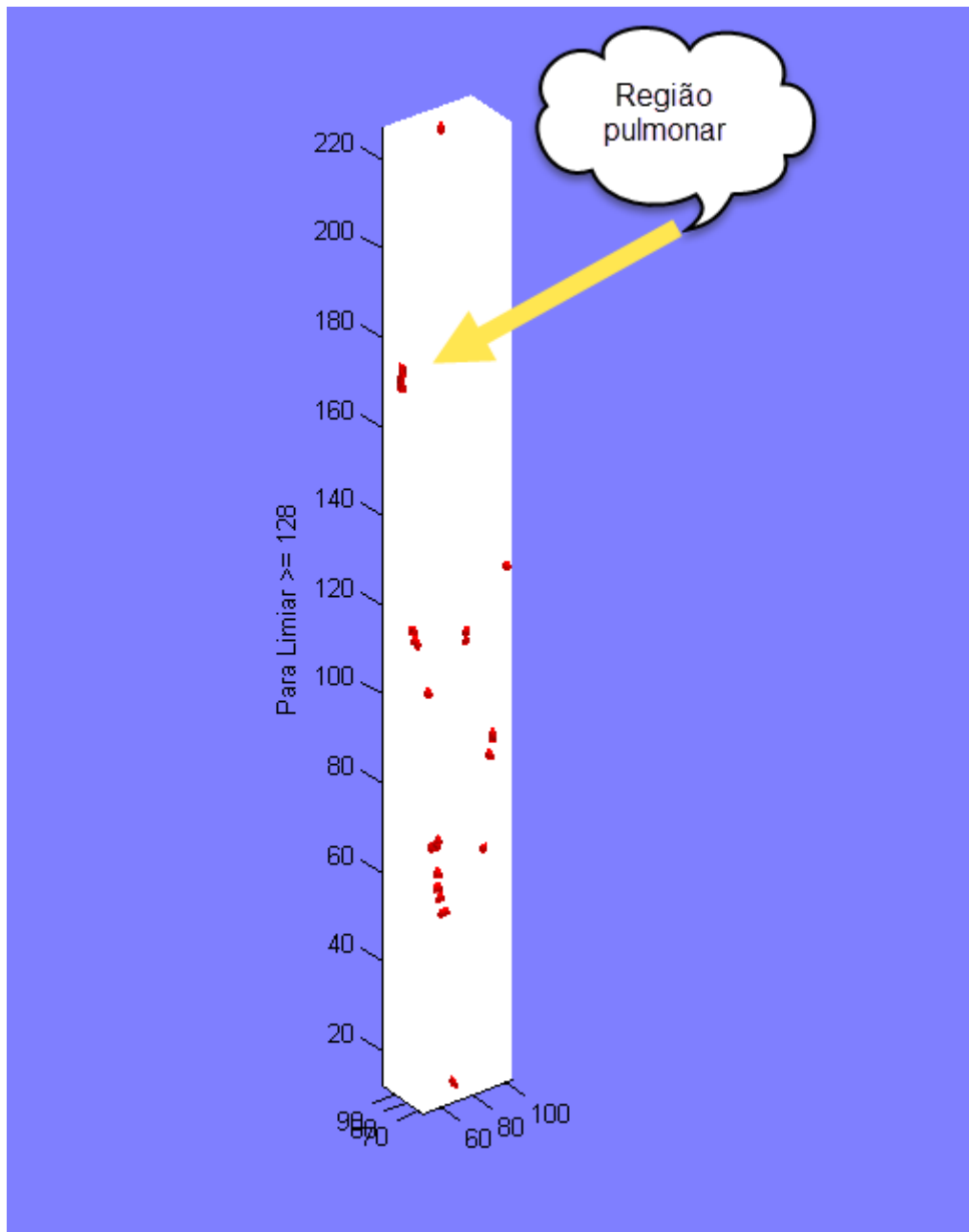


Figura 45: Pausa no limiar 128 do volume segmentado

Na figura nº 46, de vídeo podemos visualizar na pausa do plano nº 174 o único objeto de elevado contraste, a cor encarnada significa elevada intensidade, se existissem outros objetos, tomariam a cor verde média intensidade e azul baixa intensidade.

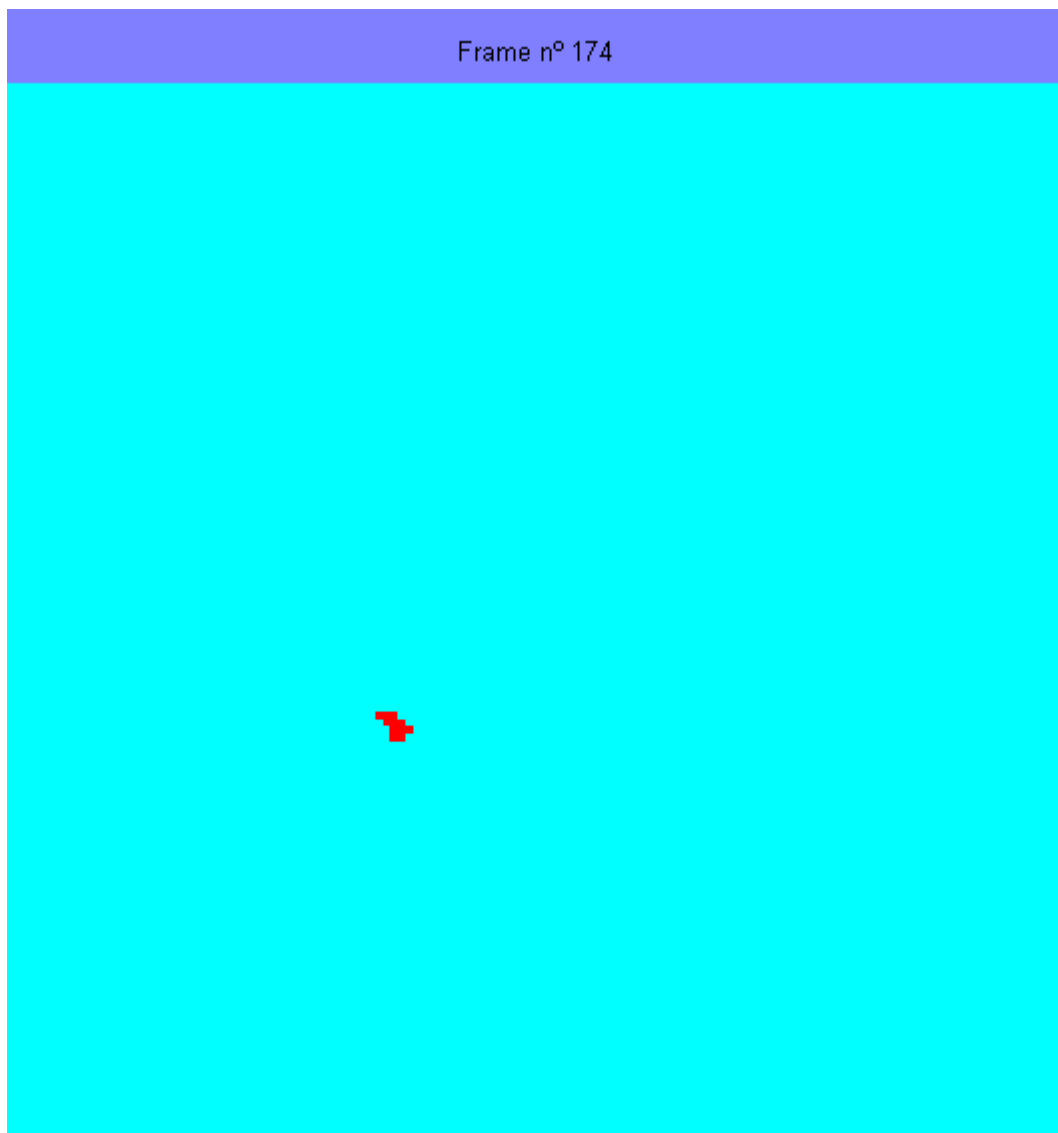


Figura 46: Objeto detetado na pausa do plano nº 174

Capítulo 8 – Conclusões gerais e Trabalhos Futuros

8.1. Síntese do Trabalho e Conclusões Gerais

Relativamente ao desenvolvimento do algoritmo para segmentar diferentes regiões no mesmo plano não foi uma tarefa trivial.

Houve algumas dificuldades iniciais na adaptação a certa sintaxe do *Matlab*, mas foi ultrapassada rapidamente.

Inicialmente era unicamente proposto encontrar regiões em imagens PET, dado que com o decorrer do tempo e o gosto pelos desafios e a atitude de me encontrar com o projeto que estava a desenvolver, teve uma evolução fantástica de tal modo que cheguei ao ponto de criação de vídeo para demonstrar a vários modos de interpretar o conceito de objeto de elevado contraste.

Posso também considerar um grande desafio foi a fazer testes a volumes de matrizes com 255 planos, onde se percebeu realmente que poderia criar novas funcionalidades ao programa, assim os cortes desde axial, coronal e sagital.

Outro desafio final prendeu-se na construção de uma interface para permitir visualizar tanto a segmentação bidimensional como a de volumes de imagens fantoma e humana.

Para as imagens humanas a função de atenuação (imagem fantoma típico igual a 1 e máximo de 4) foi ajustada para um típico igual a 3 e um máximo igual a 5, (ver função MTF figura nº 8), visto que já podíamos encontrar regiões de maior dimensão, logo tivemos de alargar a escala.

8.2. Prosseguimento de Trabalhos Futuros

Um dos trabalhos futuros a meu ver seria implementar este algoritmo em outra linguagem, para ser testado de forma a verificar a rapidez.

Se tivermos em conta a atual tecnologia, devemos proteger a informação que é gerada dentro do *software*, a segurança desta informação consiste em manter a privacidade dos dados do paciente, assegurando a sua integridade, não deixando ser afetada pelo uso de pessoas que se encontrem dentro da infra-estrutura circundante.

Como este protótipo foi desenvolvido essencialmente para testes onde unicamente o que nos permite guardar é uma folha em *excel*, mais tarde deveria ser usada uma base dados que permiti se guardar dados de paciente, como imagens de PET e resultados de segmentação para diversos momentos desse mesmo paciente.

8.3. Considerações Finais

Ao longo deste trabalho apresentamos testes e resultados obtidos num fantoma e unicamente em um paciente, penso que deveria ser testado em outros pacientes similares para uma avaliação qualitativa pela observação de resultados.

De uma forma geral, este protótipo de segmentação de imagens PET oferece diversas vantagens do que era inicialmente pensadas, sendo um algoritmo bem estruturado e com estudo prévio da estrutura à qual se destinava atingir os objetivos inicialmente previstos.

A realização deste trabalho tornou se uma experiencia bastante positiva, que além de ser um desafio, motivou me para levar o projeto além.

Considero que ganhei bastante aptidão em desenvolvimento de algoritmos em *Matlab*, juntando a experiencia que já tinha em programação em outras linguagens, sendo este um trabalho de investigação em segmentação de imagem PET, consegui criar vários protótipos, que podem apresentar um grande potencial como ferramenta de segmentação e visualização de volumes de imagem PET.

Por fim, é de realçar o excelente ambiente de trabalho que por vezes de grande intensidade se verificou existir. Com a concretização dos objetivos pretendidos acabei de ficar adepto a novos paradigmas.

Referências Bibliográficas

- (2011), Anatomia Humana: Planos de secção e Termos de posição e direção. Obtido 1 de 12 de 2014. <http://adanatomia.blogspot.pt/2011/05/planos-de-seccao-e-termos-de-posicao-e.html>
- (1994), Sensor óptico para a contagem e classificação de impurezas na pasta de papel, Tese pertencente a Francisco José Simões Duarte, Mestre em Sistemas e Tecnologias da Informação na área de Sistemas e Controlo.
- (2014), Fantomas. Obtido a 2 de 12 de 2014. <http://www.medicalexpo.com/prod/biodex/thyroid-uptake-scan-gamma-probes-mobile-78099-593131.html>
- (2014), Informação para profissionais de saúde. Obtido a 11 de 12 de 2014. http://www.uc.pt/icnas/informacao_para_profissionais/infoprofess_pet
- (2014), Informação para profissionais de saúde. Obtido a 11 de 12 de 2014. http://www.uc.pt/icnas/informacao_para_doentes/infopaciente_pet
- Morais, V; Vieira, C, (2010). MATLAB 7&6 – Curso Completo, ISBN: 978-972-722-354-1
- (2013), *Image Processing Toolbox*, Obtido 10 de 10 de 2013. <http://www.mathworks.com/help/images/index.html>
- (2013), Regiões bidimensionais. Obtido a 11 de 10 de 2013. <http://www-rohan.sdsu.edu/doc/matlab/toolbox/images/regionprops.html>
- (2013), *Techniques for Visualizing Scalar Volume Data*. Obtido a 11 de 10 de 2013. <http://www.mathworks.com/help/matlab/visualize/techniques-for-visualizing-scalar-volume-data.html>
- (2013), *Uitab customizations*. Obtido a 4 de 6 de 2013. <http://undocumentedmatlab.com/blog/uitab-customizations/>
- (2012), *Matlab Progressbar*. Obtido a 7 de 12 de 2012. <http://toto-share.com/2011/10/new-matlab-progressbar/>
- (2013), *Dicom Read Volume*. Obtido a 10 de 10 de 2013. <http://www.mathworks.com/matlabcentral/fileexchange/23237-read-and-write-single-file-dicom-volumes/content/dicomreadvolume.m>
- (2012), *Exporting to Excel*. Obtido a 2 de 4 de 2013. http://www.mathworks.com/help/matlab/import_export/exporting-to-excel-spreadsheets.html
- (2014), SUV. Obtido a 2 de 12 de 2014. <http://www.wseas.us/e-library/conferences/2010/Taipei/AIBE/AIBE-68.pdf>

Anexos

Segue em anexo a função mais importante, que trata grande parte dados, até estarem devidamente concluídos. O código está indentado e comentado, para o leitor ter uma percepção do tratamento das regiões de um plano de PET.

```
function Trata_Regiao(xi,yi)
%Variáveis Globais
global Ibinarizada_uint8;
global Valor_medio_objetos_limiar_uint8;
global Valor_medio_objetos_uint8;
global Original_uint8;
global IvmObjetoscopy;
global IMClmiar;
global limiar;
global x;
global y;
global nl;
global nc;
global In_double_uint8;
global ContrasteDasMedias;
global VerificaVizinhos;
global IBestContraste;

%Inicialização de variáveis
numPixels = 0;
SomaRegion = 0;

%Inicialização de variáveis
numPixelsVizinhos = 0;
SomaPixelsVizinhanca = 0;

%Inicialização de variáveis
I_pilha = 1;
I_pilha2 = 0;
I_pilha3 = 0;

%Declara coordenadas de entrada na pilha
pilha_x(I_pilha)= xi;
pilha_y(I_pilha)= yi;

%Ciclo While para percorrer objeto
while( I_pilha > 0)

    %Decrementa coordenadas de objeto até haver coordenadas
    x = pilha_x(I_pilha);
    y = pilha_y(I_pilha);
    I_pilha = I_pilha - 1;

    %Ciclo if para todos valores maiores que zero
    if( Ibinarizada_uint8(x,y)> 0)
        SomaRegion = SomaRegion + Ibinarizada_uint8(x,y);

        %Guarda coordenada na pilha2
        I_pilha2 = I_pilha2 + 1;
        pilha2_x(I_pilha2)= x;
        pilha2_y(I_pilha2)= y;
```

```
%Guarda coordenada na pilha2
I_pilha3 = I_pilha3 + 1;
pilha3_x(I_pilha3)= x;
pilha3_y(I_pilha3)= y;

%Incrementa um pixel ao passar aqui
numPixels = numPixels + 1;

%Coloca coordenada a zero
Ibinarizada_uint8(x,y)= 0;

end
%Verifica se existe vizinhos à volta da coordenada f(x,y) e se houver
%guarda coordenada na pilha
if(Ibinarizada_uint8(x-1,y-1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x-1 ;
    pilha_y(I_pilha) = y-1;
end

if(Ibinarizada_uint8(x-1,y)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x-1 ;
    pilha_y(I_pilha) = y;
end

if(Ibinarizada_uint8(x-1,y+1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x-1 ;
    pilha_y(I_pilha) = y+1;
end

if(Ibinarizada_uint8(x,y-1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x;
    pilha_y(I_pilha) = y-1;
end

if(Ibinarizada_uint8(x,y+1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x;
    pilha_y(I_pilha) = y+1;
end

if(Ibinarizada_uint8(x+1,y-1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x+1;
    pilha_y(I_pilha) = y-1;
end

if(Ibinarizada_uint8(x+1,y)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x+1;
    pilha_y(I_pilha) = y;
end

if(Ibinarizada_uint8(x+1,y+1)>0)
    I_pilha = I_pilha + 1;
    pilha_x(I_pilha) = x+1 ;
    pilha_y(I_pilha) = y+1;
end
```



```

end % end while

%Calcula a media do valor pixeis encontrados no limiar
Media = double(SomaRegion/numPixels);

%Calcula o modulo do contraste no limiar
Contraste = abs(limiar-Media);

%Ciclo while para colocar valores de media e contraste do limiar na matriz
while (I_pilha2>0)

    y = pilha2_y(I_pilha2);
    x = pilha2_x(I_pilha2);
    I_pilha2 = I_pilha2 - 1;

    Valor_medio_objetos_limiar_uint8(x,y)= Media;
    IvmObjetoscopy(x,y) = Contraste;

end% while pilha2

%Inicio de verificação de media no limiar
copia = 1;
for ip = 1:I_pilha3

    if IMClimiar(pilha3_x(ip),pilha3_y(ip))>
IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip));

        copia = 0;

    end

    if VerificaVizinhos(pilha3_x(ip),pilha3_y(ip))>
IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip));

        copia = 0;

    end

end

if copia > 0
    %Copia valores para as coordenadas
    for ip = 1:I_pilha3

        IMClimiar(pilha3_x(ip),pilha3_y(ip))=
IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip));
        VerificaVizinhos(pilha3_x(ip),pilha3_y(ip))=
IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip));
        VerificaVizinhos(pilha3_x(ip),pilha3_y(ip))=
In_double_uint8(pilha3_x(ip),pilha3_y(ip));
        IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip))=0;

    end
end

```

```

else
    %Repõe valores da matriz a zero
    for ip = 1:I_pilha3

        IvmObjetoscopy(pilha3_x(ip),pilha3_y(ip))=0;

    end

end % end if >0

%Detectar pontos vizinhos do objeto
for x = 1:nl
    for y = 1:nc

        if (VerificaVizinhos(x,y)>1)
            VerificaVizinhos(x,y)=0;
            if (VerificaVizinhos(x-1,y-1)==0)
                if (In_double_uint8(x-1,y-1)>=limiar)
                    VerificaVizinhos(x-1,y-1) = In_double_uint8(x-1,y-1);

                    SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x-1,y-1);

                    numPixelsVizinhos = numPixelsVizinhos + 1;
                    VerificaVizinhos(x-1,y-1)= 1;

                end

            end
            if (VerificaVizinhos(x-1,y)==0)
                if (In_double_uint8(x-1,y)>=limiar)
                    VerificaVizinhos(x-1,y) = In_double_uint8(x-1,y);

                    SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x-1,y);

                    numPixelsVizinhos = numPixelsVizinhos + 1;
                    VerificaVizinhos(x-1,y)= 1;

                end

            end
            if (VerificaVizinhos(x-1,y+1)==0)
                if (In_double_uint8(x-1,y+1)>=limiar)
                    VerificaVizinhos(x-1,y+1) = In_double_uint8(x-1,y+1);

                    SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x-1,y+1);

                    numPixelsVizinhos = numPixelsVizinhos + 1;
                    VerificaVizinhos(x-1,y+1)= 1;

                end

            end
            if (VerificaVizinhos(x,y-1)==0)
                if (In_double_uint8(x,y-1)>=limiar)
                    VerificaVizinhos(x,y-1) = In_double_uint8(x,y-1);

```

```

        SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x,y-1);

        numPixelsVizinhos = numPixelsVizinhos + 1;
        VerificaVizinhos(x,y-1)= 1;

    end
end
if(VerificaVizinhos(x,y+1)==0)
    if(In_double_uint8(x,y+1)>=limiar)
        VerificaVizinhos(x,y+1) = In_double_uint8(x,y+1);

        SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x,y+1);

        numPixelsVizinhos = numPixelsVizinhos + 1;
        VerificaVizinhos(x,y+1)= 1;

    end
end
if(VerificaVizinhos(x+1,y-1)==0)
    if(In_double_uint8(x+1,y-1)>=limiar)
        VerificaVizinhos(x+1,y-1) = In_double_uint8(x+1,y-1);

        SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x+1,y-1);

        numPixelsVizinhos = numPixelsVizinhos + 1;
        VerificaVizinhos(x+1,y-1)= 1;

    end
end
if(VerificaVizinhos(x+1,y)==0)
    if(In_double_uint8(x+1,y)>=limiar)
        VerificaVizinhos(x+1,y) = In_double_uint8(x+1,y);

        SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x+1,y);

        numPixelsVizinhos = numPixelsVizinhos + 1;
        VerificaVizinhos(x+1,y)= 1;

    end
end
if(VerificaVizinhos(x+1,y+1)==0)
    if(In_double_uint8(x+1,y+1)>=limiar)
        VerificaVizinhos(x+1,y+1) = In_double_uint8(x+1,y+1);

        SomaPixelsVizinhanca = SomaPixelsVizinhanca +
In_double_uint8(x+1,y+1);

        numPixelsVizinhos = numPixelsVizinhos + 1;
        VerificaVizinhos(x+1,y+1)= 1;

    end
end

end% end if ==0

```

```

        end
    end

    %Calculos da Média dos vizinhos
    MediaVizinhanca = uint8(SomaPixelsVizinhanca/numPixelsVizinhos);

    %Calculo do modulo da media do objeto e da sua vizinhança
    ContrasteMedias = abs( MediaVizinhanca - Media);
    %Calculo da atenuação do objeto
    ContrasteMedias = ContrasteMedias*Mtf(numPixels);

    %Ciclo percorre a matriz e coloca os valores do ContrasteMedias
    for ip = 1:I_pilha3

        ContrasteDasMedias(pilha3_x(ip),pilha3_y(ip))= ContrasteMedias;

    end

    %Inicio da verificação da copia de melhor diferença do ContrasteMedias
    copia2 = 1;
    for ip = 1:I_pilha3

        if IBestContraste(pilha3_x(ip),pilha3_y(ip))>
            ContrasteDasMedias(pilha3_x(ip),pilha3_y(ip));

            copia2 = 0;

        end

    end

    if copia2 > 0

        for ip = 1:I_pilha3

            IBestContraste(pilha3_x(ip),pilha3_y(ip))=
            ContrasteDasMedias(pilha3_x(ip),pilha3_y(ip));
            Valor_medio_objetos_uint8(pilha3_x(ip),pilha3_y(ip))=
            ContrasteDasMedias(pilha3_x(ip),pilha3_y(ip));
            VerificaVizinhos(pilha3_x(ip),pilha3_y(ip))=0;

        end

    else

        for ip = 1:I_pilha3

            ContrasteDasMedias(pilha3_x(ip),pilha3_y(ip))=0;

        end

    end % end if >0
end

```

```
%Coloca valores originais calculo do valor medio de acordo com I_pilha3
for x = 1:nl
    for y = 1:nc

        if Valor_medio_objetos_uint8(x,y) > 0
            Valor_medio_objetos_uint8(x,y) = Original_uint8(x,y);

        end

    end
end

end%%% end function
```

